НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Кафедра обчислювальної техніки

«На правах рукопису»
УДК 004.052.42

До захисту допущено:
Завідувач кафедри
_____ *Сергій СТІРЕНКО*
«___»_____2021 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-науковою програмою «Комп'ютерні системи та мережі»

зі спеціальності 123 «Комп'ютерна інженерія»

на тему: «Метод та засоби відновлення даних в системах їх віддаленого зберігання з використанням лінійних надлишкових кодів»

Виконала:
студентка VI курсу, групи ІВ-91мн
Дорошенко Лариса Юріївна _____

Керівник:
доцент, к.т.н., доцент,
Марковський Олександр Петрович _____

Консультант з нормоконтролю:
професор, д.т.н., професор,
Кулаков Юрій Олексійович _____

Рецензент:
декан ФПМ, д.т.н, професор,
Дичка Іван Андрійович _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2021 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет (інститут) *Інформатики та обчислювальної техніки*
(повна назва)

Кафедра *Обчислювальної техніки*
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-науковою програмою
Спеціальність *123. Комп'ютерна інженерія*
(код і назва)

Спеціалізація *123. Комп'ютерні системи та мережі*
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Стіренко С.Г.
(підпис) (ініціали, прізвище)
« » 20***21*** р.

ЗАВДАННЯ
на магістерську дисертацію студентки
Дорошенко Ларисі Юріївні
(прізвище, ім'я, по батькові)

1. Тема дисертації Метод та засоби відновлення даних в системах їх віддаленого зберігання з використанням лінійних надлишкових кодів

Науковий керівник дисертації доц.,к.т.н., доц. Марковський О.П.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «12» березня *2021* р. №809-с

2. Строк подання студентом дисертації *26 квітня 2021 р.*

3. Об'єкт дослідження *процеси резервування та відновлення інформаційних блоків при їх рознесеному зберіганні на віддалених сховищах з використанням хмарних технологій*

4. Предмет дослідження *методи формування резервних блоків і методи відновлення втрачених інформаційних блоків з використанням резервних даних*

5. Перелік завдань, які потрібно розробити: *аналіз особливостей сучасного та перспективного стану розподіленого віддаленого зберігання даних користувачів*

з використанням хмарних технологій; розробка методу відновлення втрачених даних; розробка способу прискореного відновлення втрачених при віддаленому зберіганні блоків даних шляхом використання спеціальних таблиць; розробка програмних засобів для реалізації розробленого методу; теоретичне та експериментальне дослідження ефективності розробленого методу

6. Консультанти розділів дисертації:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Кулаков Ю.А., професор		

7. Дата видачі завдання 26.11.2020

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів дисертації	Примітка
1.	*Затвердження теми роботи*	*10.12.2020-15.12.2020*	
2.	*Вивчення та аналіз завдання*	*15.12.2020-31.01.2021*	
3.	*Розробка архітектури та загальної структури системи*	*01.02.2021-10.02.2021*	
4.	*Розробка структур окремих підсистем*	*10.02.2021-20.02.2021*	
5.	*Програмна реалізація системи*	*20.02.2021-10.04.2021*	
6.	*Оформлення пояснювальної записки*	*10.04.2021-25.04.2021*	
7.	*Передзахист*	*26.04.2021*	
8.	*Захист*	*17.05.2021*	

Студент _____ Л.Ю. Дорошенко
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ О.П. Марковський
(підпис) (ініціали, прізвище)

РЕФЕРАТ

на магістерську дисертацію

виконану на тему: Метод та засоби відновлення даних в системах їх віддаленого зберігання з використанням лінійних надлишкових кодів»

студенткою: Дорошенко Ларисою Юріївною

Робота складається із вступу та 4 розділів. Сумарний об’єм роботи: 89 аркушів тексту, 9 ілюстрацій та 3 таблиці. Для магістерської дисертації було використано літературу з 65 різноманітних джерел.

Актуальність. Поява хмарних технологій докорінно змінила способи довготривалого зберігання і розповсюдження інформації.

Надійність є невід’ємним елементом збереження даних на віддалених носіях. Для підвищення надійності інформацію розбивають на частини та зберігають на різних віддалених носіях. Відновлення втраченої інформації відбувається за рахунок неушкоджених частин даних. Одним із відомих методів є резервування. Ефективність резервування досягається балансом між надійністю збереження даних, швидкістю їх відновлення та об’ємом резервної інформації.

Отже, задача підвищення ефективності відновлення втраченої інформації, що зберігається на віддалених носіях, та створення методу резервування даних, що відповідають сучасним вимогам, є актуальною з сучасним розвитком хмарних технологій та за поширеного використання сервісів віддаленого збереження інформації.

Мета і завдання дослідження. Мета магістерської роботи полягає в підвищенні ефективності відновлення блоків даних за рахунок урахування їх важливості, шляхом збільшення ймовірності відновлення більш важливих блоків за умови, що кількість втрачених блоків перевищує поріг гарантованого відновлення.

Для досягнення поставленої мети в магістерські дисертації вирішуються такі задачі:

1. Аналіз особливостей сучасного та перспективного стану розподіленого віддаленого зберігання даних користувачів з використанням хмарних технологій, обґрунтування критеріїв ефективності відновлення втраченої інформації, яка зберігається на віддалених сховищах. Аналітичний огляд існуючих методів резервування та відновлення втрачених даних, виявлення можливостей підвищення їх ефективності, а також визначення напрямків досліджень.

2. Розробка методу відновлення втрачених даних, який відрізняється від відомих тим, що дозволяє враховувати інформаційну важливість окремих блоків даних шляхом побудови матриці формування резервних блоків.

3. Розробка способу прискореного відновлення втрачених при віддаленому зберіганні блоків даних шляхом використання спеціальних таблиць.

4. Розробка програмних засобів для реалізації розробленого методу резервування та відновлення втрачених при віддаленому зберіганні блоків даних, а також імітаційного та статистичного моделювання його функціонування.

5. Теоретичне та експериментальне дослідження ефективності розробленого методу.

Об’єкт дослідження – процеси резервування та відновлення інформаційних блоків при їх рознесеному зберіганні на віддалених сховищах з використанням хмарних технологій.

Предмет дослідження – методи формування резервних блоків і методи відновлення втрачених інформаційних блоків з використанням резервних даних.

Методи досліджень базуються на теорії ймовірності та математичної статистики, теорії лінійних кодів, методах організації резервування, а також на використанні методів імітаційного та статистичного моделювання.

Наукова новизна одержаних результатів роботи. Розроблено метод відновлення втрачених при віддаленому зберіганні даних, який відрізняється від відомих тим, що дозволяє враховувати інформаційну важливість окремих блоків даних шляхом побудови матриці формування резервних блоків з використанням пріоритетів, що забезпечує підвищення ймовірності відновлення більш важливих

блоків при зменшенні ймовірності реконструювання менш важливих за умови, що кількість втрачених блоків перевищує поріг гарантованого відновлення.

Практична значимість результатів дослідження визначається тим, що воно направлено на урахування особливостей реальних систем віддаленого зберігання інформації в хмарах з урахуванням важливості даних. Управління резервуванням даних здійснюється на рівні користувача, що дозволяє адаптувати систему резервування та відновлення до потреб користувача. В комплексі з спеціальною організацією розміщення даних користувачів на сховищах та моніторингом доступності даних, застосування цього методу дозволяє звести практично до нуля ймовірність їх втрати.

Апробація результатів дисертації. Основні результати дисертації доповідались та обговорювались на 4-х міжнародних науково-технічних конференціях:

1. Міжнародна наукова конференція “Security, Fault Tolerance, Intelligence: ICSFTI2019”. м.Київ, 14-15 травня 2019 р.

2. Міжнародна наукова конференція “Science and Technology of the XXI Century” м.Київ, 17 грудня 2020 р.

3. ІІ-га Міжнародна науково-практична конференція “Наука та концепції”. м.Київ, 29-30 квітня 2019 р.

4. ІІІ-тя Міжнародна науково-практична конференція “Наука та концепції”. м.Київ, 14-15 травня 2020 р.

Публікації. Основні положення магістерської дисертації опубліковані в 4 наукових працях, серед яких всі – матеріали наукових конференцій.

1. Doroshenko L. Organization of Reservation and Reconstruction of Data / Doroshenko L., Markovskyi O., Honchar A. // International Conference ICSFTI2019 (Kyiv, May 14−15, 2019). Kyiv, 2019. P. 29-34.

2. Doroshenko L. Organization of Data Recovery when Storing in the “Clouds” / Doroshenko L. // Proceedings of the XXI International Students R&D Online Conference (Kyiv, 17 December, 2020). Kyiv, 2020. P. 29-31.

3. Дорошенко Л.Ю. Метод резервування даних з використанням лінійних відновлюючих кодів / Л.Ю. Дорошенко, В.Ю. Куц // Матеріали II міжнарод. наук.-практ. конф. Наука та концепції: (м. Київ, 29-30 квіт. 2019 р.). Київ, 2019. С. 19-22.

4. Дорошенко Л.Ю. Організація відновлення даних при їх зберіганні в хмарах / Л.Ю. Дорошенко, О.В. Русанова, В.Ю. Куц // Матеріали III міжнарод. наук.-практ. конф. (м. Київ, 14-15 трав. 2020 р.). Київ, 2020. С. 33-37.

Ключові слова

лінійні надлишкові коди, віддалені носії, відновлення блоків даних, erasure codes, резервування даних

ЗМІСТ

ВСТУП

Поступальний розвиток технології комп’ютерної обробки інформації нерозривно пов’язаний з проблемою забезпечення високого рівня надійності всіх компонентів обчислювальних систем, в тому числі засобів зберігання даних. Високі темпи зростання можливостей сучасних магнітних і твердотільних накопичувачів та прогрес мережових технологій дозволили в останнє десятиліття вийти на якісно новий рівень розвитку комп’ютерних систем, визначальною особливістю якого стала можливість віддаленого надання практично необмежених ресурсів пам'яті широкому колу користувачів в рамках хмарних технологій. Розподілені системи зберігання інформації дозволяють значно розширити можливості ефективного використання комп’ютерних технологій в усіх сферах людської діяльності, суттєво підвищити надійність довготривалого зберігання даних користувачів. Висока надійність зберігання даних користувачів на віддалених рознесених сховищах зумовлена обмеженістю дії чинника, що призводить до втрати інформації. За цих умов об’єктивно зростає значення ефективного застосування технологій резервування та відновлення даних.

Динамічний розвиток таких систем не можливий без забезпечення високого рівня надійності доступу кожного користувача до своїх даних, що зберігаються на віддалених від нього і не підконтрольних йому носіях. Це вимагає створення нових, адекватних розвитку систем розподіленого зберігання інформації, методів та засобів відновлення даних, до яких втрачено доступ.

За умови швидкого зростання об’ємів такої інформації і розширення кола користувачів технологій віддаленого зберігання даних набуває ваги задача підвищення ефективності методів та засобів резервування та відновлення інформації в рамках таких технологій. Це вимагає постійного пошуку нових підходів, методів та розробки засобів забезпечення неперервності доступу до даних шляхом їх відновлення з використанням резервної інформації. З цих позицій наукова задача підвищення ефективності резервування та відновлення

даних в розподілених системах віддаленого зберігання інформації є актуальною для сучасного етапу розвитку розподілених комп’ютерних систем.

РОЗДІЛ 1

АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА ЗАСОБІВ ВІДНОВЛЕННЯ ВТРАЧЕНИХ ПРИ ВІДДАЛЕНОМУ ЗБЕРІГАННІ ДАНИХ

1.1. Постановка задачі відновлення втрачених даних при їх віддаленому зберіганні

Розвиток глобальних мереж останніх десятиліть вніс суттєві зміни в процес комп'ютерного опрацювання інформації в сторону розподілених систем для збереження інформації та віддалених обрахунків. Розповсюдженого застосування отримало розподілене збереження даних клієнтів на ремоуті [2]. Наразі технологія збереження даних у клауді надає ряд переваг (рис. 1.1):

- можливість доступу до даних клієнта для інших клієнтів;
- практично відсутній ліміт на об'єм пам'яті;
- існують зручні інструменти для обміну інформацією;
- якісно вища степінь забезпечення надійності;

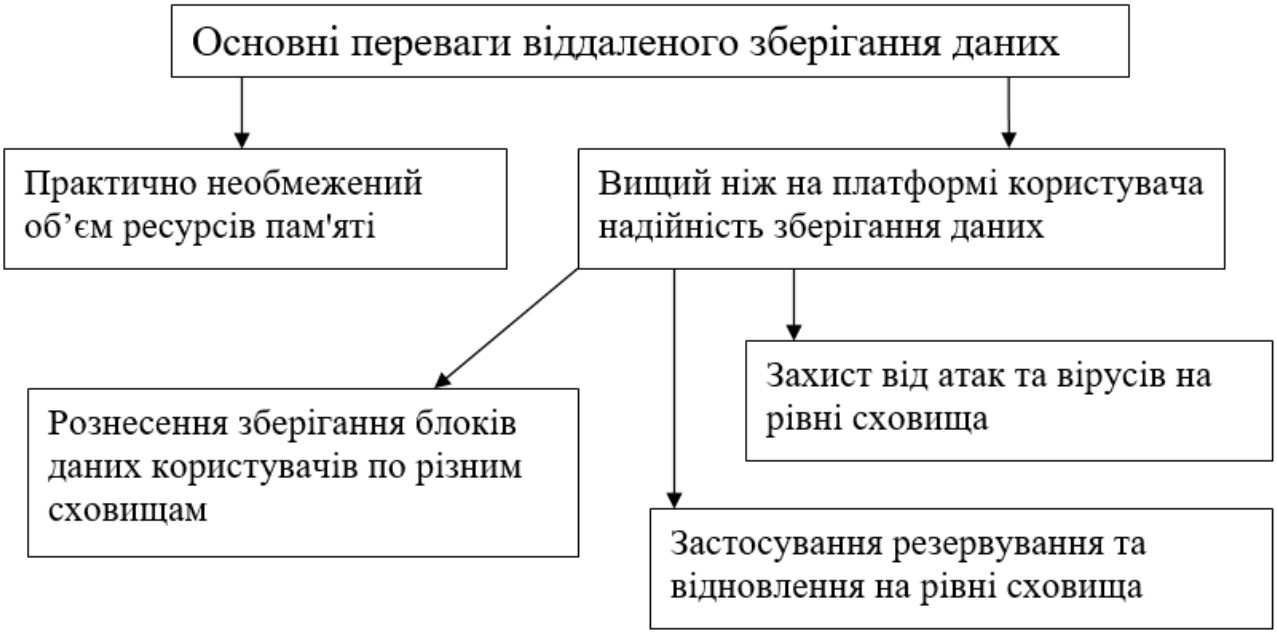

Рис.1.1. Головні переваги зберігання інформації на віддалених носіях

Інформація клієнта, що зберігається або на індивідуальних носіях, або на носіях віддаленого збереження, може втратитися внаслідок певних факторів, їх відображено на рис. 1.2.

Рис.1.2. Причини втрати при віддаленому зберіганні даних

Причинами втрати інформації у випадку її збереження на магнітних, оптичних носіях або флеш-пам’яті можуть стати такі причини як: вихід з ладу обладнання, помилки роботи ПЗ (програмного забезпечення), злочинні атаки зі сторони хакерів, вірусні атаки, катаклізми, що мають політичний, терористичний, природний, військовий або техногенний характер [3].

У таких випадках можлива втрата декількох або всіх блоків інформації на деякому індивідуальному або віддаленому сховищі. За умови зберігання інформації клієнта на ремоуті головним фактор для забезпечення надійності є поділ та збереження даних клієнта на окремих, територіально віддалених

сховищах [4]. Таким чином можна обмежити втрати до порівняно невеликих об’ємів внаслідок дії вищезазначених чинників.

Проте, крім втрати для клієнта може тимчасово бути відсутній доступ до віддаленого сховища через вихід обладнання з ладу, профілактичні роботи, перезавантаження мережі або сховища.

Не дивлячись на ці фактори, кількість клієнтів, що користуються віддаленими сховищами для збереження інформації, зростає, оскільки збереження інформації на ремоуті гарантує більш високу надійність [5].

Менша вірогідність втрати даних, можливість використання великих обсягів пам’яті, простота в поширенні та обміні інформаціє є причинами стрімкого поширення збереження даних на ремоуті у клауді. Щоб забезпечити збільшення надійності збереження даних вони розділяються та зберігаються на географічно різних вузлах. Існує надзвичайно низька ймовірність того, що одночасно на всіх вузлах вищенаведені фактори вплинуть на втрату інформації. Завдяки цьому лімітується обсяг втрачених даних, що можна відновити за рахунок резервного копіювання [6].

Всередині одного окремого вузла впорядковано поділ інформації за носіями, клієнтський доступ до неї, захисні інструменти та засоби, виконання резервного копіювання для випадку зникнення доступу клієнта до одного або кількох носіїв.

Зрозуміло, що виконання резервного копіювання на рівні одного вузлу не дозволить суттєво знизити вірогідність втрати даних, що спричинено катаклізмами, зловмисними атаками хакерів або через збій роботи ПЗ.

Наразі стрімкий розвиток мережі Інтернет робить можливим виконувати резервне копіювання на рівні клієнта чи групи клієнтів. Одночасно з цим, щоб локалізувати можливі втрати інформації, можна застосувати розподіл збереженої інформації. Лише згаданий фактор може дійсно попередити втрату інформації через катаклізми. Але потрібно врахувати, що збереження інформації на різних носіях може негативно відобразитися на те, як швидко клієнт може

доступитися до неї, оскільки є необхідність зібрати її разом з різних репозиторіїв [7].

Виходячи з цього, можна сказати, що для того, щоб підвищити надійність збереження клієнтської інформації враховуючи згадані спеціалізовані вимоги, потрібно виконувати резервне копіювання на рівні більш високу за рівень вузла [8]. Критерії для ефективного виконання резервного копіювання та відновлення інформації при збереженні на ремоуті зображені на рис. 1.3.

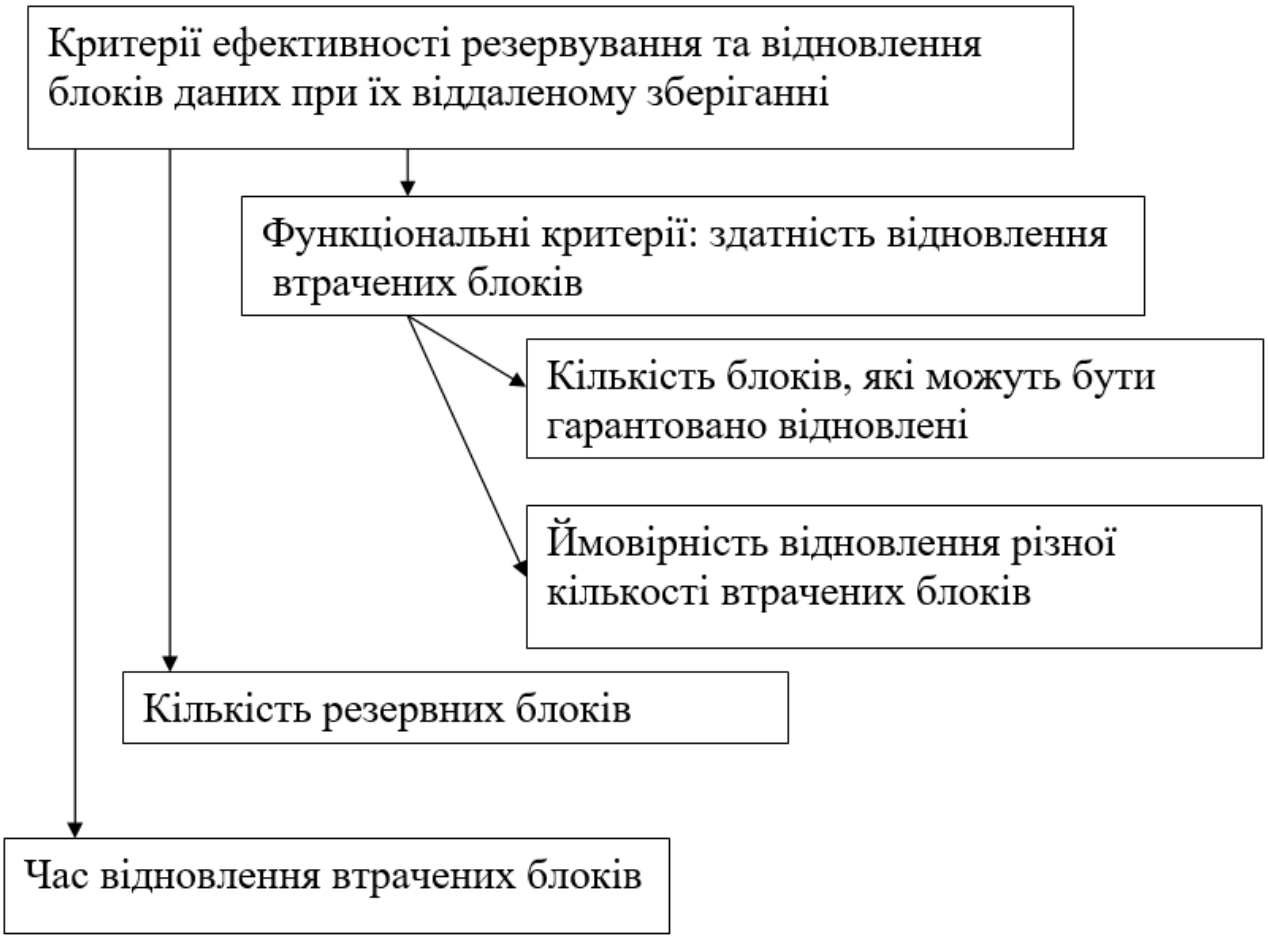

Рис.1.3. Критерії для ефективного виконання резервного копіювання та відновлення інформації при збереженні на ремоуті

Критерії, що зазначені вище, протирічать між собою, з цього фактор ефективності береться як ступінь віднаходження компромісного рішення між згаданими критеріями та важливістю обраного критерію, що визначається специфікою використання.

Щоб поставити ціль дослідження необхідно стисло дати характеристику вже наявним системам резервного копіювання та відновлення інформації.

Новим системам збереження даним притаманна багаторівнева структура для організації інструментів, що забезпечують надійність. Наприклад, при збереженні інформації за допомогою магнітних або оптичних дисків, то 1-ий рівень формують коди корегуючі (ECC – error correcting code) для кожного сектору [9]. Відповідно до наявних стандартів вони являють собою 4 контрольні байти, що доповнюють 512 байтів інформації сектору. Їх використання дозволяє виконати локалізацію та виправлення будь-яких 2-ох байт серед 512 байтів інформації сектору.

Для того, щоб забезпечити рівень надійності збереження інформації формування 1-ого рівня виконується за допомогою засобів, які працюють на рівні секторів, що на магнітних чи оптичних носіях. У випадку, коли їх форматують, виконується контрольне читання даних. Якщо знайшли два або більше спотворені байти, то сектор маркується як дефектний і відбувається блокування запису даних до нього. Додатково до 512 інформаційних байтів сектор містить 4 байти резервного копіювання — контрольні символи Ріда-Соломона [11], за допомогою яких можна виправити спотворені два байти або при спотворенні більшого числа байт виявити сам факт спотворення. Отже, 1-ий рівень включає класичні корегуючі коди [12], що виконують виправлення спотворених байтів у два етапи:

- локалізація символів, що були спотворені,
- відновлення цих символів.

Другий рівень для сховищ віддаленого збереження клієнтської інформації, щоб забезпечити надійність, формується з використанням накопичувачів резервного копіювання. Загально відомою технологією другого рівня являється технологія (Redundant Array of Independent Disks) [13].

1.2. Огляд та аналіз існуючих засобів відновлення втрачених даних

Велика кількість наявні засоби для відновлення втраченої інформації ґрунтуються на використанні MDS-кодів. За їх застосування як відновлюючих кодів в розподілених системах збереження інформації можливо досягнути найбільшу вірогідність відновлення втраченої інформації враховуючи вказаний параметр надлишковості пам'яті [50].

Практично, на MDS-кодах ґрунтуються всі відмовостійкі блочні коди. Вони виконують відображення блоку з інформацією довжиною k біт у кодове слово розрядністю n біт, причому $n > k$, із застосуванням $n - k$ надлишкових бітів. Блокові коди будуть лінійними за умови, що сума по модулю 2 будь-якої підмножини кодових слів входить у множину кодових слів [41].

Приклад. Нехай утворююча матриця G, де $n = 6$ та $k = 3$ буде мати такий вид:

$$G = \begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{vmatrix},$$

тоді значення кодової відстані між кодовими словами рівне 3.

Щоб закодувати інформаційне слово, яке представлене бінарним вектором $a = \{a_1, a_2, \dots, a_k\}$, необхідно виконати множення цього вектора та утворюючої матриці G лінійного коду: $\bar{b} = \bar{a} * G$. Тоді отримуємо n-розрядний вектор $\bar{b} = \{b_1, b_2, \dots, b_n\}$, який формується додаванням контрольних бітів до інформаційного слова.

Приклад. Застосуємо ту саму утворюючу матрицю G (1.1), тоді будемо мати кодове слово $b = 101\ 101$ як кодування для інформаційного слова $a = 101$.

Щоб декодувати блокові лінійні коди, виконується утворення перевірочної матриці H, яка утворюється за формулою:

$$H = \left|\frac{P^T}{I_{n-k}}\right|,$$

де I_{n-k} – одинична матриця розміром $n - k$,

P^T – транспонована підматриця P матриці $G = \left|\frac{I_k}{P_{n-k}}\right|$.

Тоді підматриці P та P^T, якщо візьмемо утворюючу матрицю G (1.1) будуть такі:

$$P = \begin{vmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{vmatrix} \qquad P^T = \begin{vmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{vmatrix}.$$

Тоді перевірочна матриця H буде відповідно такою:

$$H = \begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{vmatrix}$$

Відповідні рядки обох матриць G і H є ортогональними між собою. З цього випливає така умова [41]:

$$G * H^T = 0. \tag{1.1}$$

Тоді з умови (1.1) можна зробити висновок, що для кожного кодового слова b, що є результатом множення a*G буде виконуватися умова ортогональності: $\bar{b} * H^T = \bar{a} * G * H^T = 0$. За випадку, коли при передачі або при збереженні кодове слово спотворене, до цього слова додають вектор помилки $\bar{e} = \{e_1, e_2, \dots, e_n\}$: $\bar{c} = \bar{b} \oplus \bar{e}$. Під час декодування коду c виконується обрахунок для синдрому $S = \{s_1, s_2, \dots, s_{n-k}\}$, що має вигляд $S = c * H^T$. Вектор помилки буде єдиним, від чого залежатиме код синдрому S: $S = c * H^T = e * H^T \oplus b * H^T = e * H^T$. Тому, при ситуації, коли код синдрому S має своїми складовими нульові значення, код c входить у множину дозволених кодових слів. За умови не рівності синдрому нульовому значенні, він буде містити дані вектору помилки.

Приклад. Нехай кодове слово b = 101 101, вектор помилки e = 010 000, тоді маємо c = 111 101.

Синдром $S = c * H^T = e * H^T = 010$.

Всім можливим кодам синдромів можна однозначно представити код помилки, а він може однозначно локалізувати символ, що було спотворено, у кодовій комбінації. Наведемо приклад синдромів, щоб виправити однократні помилки використовуючи утворюючу матрицю G (1.1), результати занесено у таблицю 1.1.

Таблиця 1.1

Приклад синдромів, щоб виправити однократні помилки

Вектор помилки	100000	010000	001000	000100	000010	000001
Код синдрому	100	010	001	011	111	110

З технологічної точки зору, процес декодування вміщує формування таблиці синдромів із застосуванням транспонованої перевірочної матриці. Для всіх одержаних кодів обраховують значення синдрому як результат добутку коду та транспоновану перевірочну матрицю. Далі за обрахованим синдромом за таблицею визначається вектор помилки. Останнім кроком є доповнення одержаного коду вектором помилки, після чого маємо скорегований код.

У практичних завданнях при випадку, коли виправляються помилки за допомогою завадостійкого кодування, поширеним є застосування стандартних таблиць розміщень. Застосовуючи завчасно утворені таблиці з передобчисленнями можна збільшити швидкість за якої можна відновити спотворені символи. Поширені технології для відновлення втраченої інформації із застосуванням корегуючих кодів можуть включати наявні попередньо обраховані значення коду відновлення для кожного з варіантів спотворення блоку даних. Такі обрахунки часто поміщають у так звані стандартні таблиці розміщення [51].

У стандартній таблиці декодування організація кодових векторів за таких умов, що:

- V – лінійний (n, k)-розрядний код,
- v_0 – вектор, що складається з «0»,
- $v_1, v_2, \dots, v_{m-1}$ – інші кодові вектори при m=2^k,

відбувається за таким принципом. В 1-у колонку 1-ого рядка кладуть вектор v_0, що складається з «0». Під ним розміщують вектори, у які перетворюється вектор v_0 під час передачі за вказаної кратності помилок. Позначимо їх як $q_1, q_2, \dots, q_t$ при $t = 2^{n-k} - 1$. Далі в 1-ий рядок кладуть відповідно вектори $v_1, v_2, \dots, v_{m-1}$.

Подальше заповнення для рядків стандартної таблиці організовується таким чином: кожен з елементів таблиці β_{ij}, розташування якого – i-ий рядок, j-а позиція, де $i \in \{2,3,\dots,t\},\ j \in \{2,3,\dots,m-1\}$, знаходиться як результат операції суми по модулю 2 q_i, розташований на i-ому рядку в 1-ому стовпчику, та коду v_j, розташованому на 1-ому рядку в j-ій позиції: $\beta_{ij} = q_i \oplus v_j$. Реально, стандартна таблиця повністю відповідає таблиці декодування, тому через те, що лінійний код V представляє під собою підгрупу групи B^n, вище виконане розкладання є нічим іншим як розкладання по підгрупі V групи B^n. Слід зазначити, що в стандартній таблиці кодів вміщається значення всіх кодів, що мають розрядність n. Рядки таблиці виступають як суміжні класи, в той час як вектори 1-ого стовпчика – як утворюючі вектори суміжних класів. Тоді можна представити таблицю стандартної локалізації відповідно:

$$\begin{array}{ccccc} v_0 & v_1 & \cdots & v_{m-2} & v_{m-1} \\ q_1 & q_1 \oplus v_1 & \cdots & q_1 \oplus v_{m-2} & q_1 \oplus v_{m-1} \\ & & \vdots & & \\ q_t & q_t \oplus v_1 & \cdots & q_t \oplus v_{m-1} & q_t \oplus v_m \end{array}$$

У випадку, коли таблицю стандартного розміщення використовують як таблицю пришвидшеного декодування – прийнятий вектор із j-ого рядка таблиці перетворюється у вектор q_j, а саме – у елемент першого стовпчика даного рядка.

Приклад. Нехай утворююча матриця G, у якій n = 6 та k = 3, тоді

$$G = \begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{vmatrix}$$

і маємо таку множину векторів корекції $V = \{000\ 000, 100\ 011, 010\ 101, 001\ 110, 110\ 110, 101\ 101, 011\ 011, 111\ 000\}$. Отже таблиця стандартного розміщення буде відповідно такою:

000000	100011	010101	001110	110110	101101	011011	111000
100000	000011	110101	101110	010110	001101	111011	011000
010000	110011	000101	011110	100110	111101	001011	101000
001000	101011	011101	000110	111110	100101	010011	110000
000100	100111	010001	001010	110010	101001	011111	111100
000010	100001	010111	001100	110100	101111	011001	111010
000001	100010	010100	001101	110111	101100	011010	111001

Вагомою відмінністю між технологіями відтворення інформації, втраченої при віддаленому зберіганні або трансферу глобальною мережею, від технологій, що використовуються у випадку корекції інформації із використанням завадостійких кодів є наступною: вирішення лінійних рівнянь напряму не відновлює інформацію, а математично задає процедуру її відновлення. А безпосередньо процес відтворення включає передачу мережею блоків інформації, необхідних для відтворення загублених блоків, їх буферизацію та виконання операцій обчислення по відновленню загублених блоків [62].

Задача відтворення блоків інформації, збережених на ремоут системах, не передбачає такого вирішення, так як у випадку відновлення пакетів, до яких тимчасово чи постійно втрачено доступ, ще немає потрібної інформації для їх відтворення, оскільки вона віддалено зберігається, як і загублені блоки.

1.3. Дослідження існуючих засобів для відновлення даних

Проблема Існує необхідність у виправленні помилок при передачі даних по глобальних мережах в бездротовому каналі, схильних до впливу зовнішніх факторів (наприклад, ектромагнітні завади). З цього випливає проблема, що має практичне значення. В даний час існує кілька способів виправлення помилок при передачі символів [2].

Розділяють наступні групи критеріїв, які є характеристикою ефективності засобів виправлення помилок:

- функціональні – характеристика здатності виконувати завдання за виявленням та виправленням помилок;
- обчислення – характеристика реалізації виправлення в програмній та апаратній областях;
- інформаційні – характеристика рівню вибору додаткової (надлишкової) інформації для виправлення помилок.

Функціональні критерії включають в себе множинність помилок, при якій можливо гарантоване виправлення без необхідності повторної передачі,

вірогідність корекції помилок з множинністю, що перевищує гарантовану, множинність помилок, які гарантовано виявляться.

Обчислювальні критерії включають обчислювальну і часову складність обчислювальних операцій для виявлення та виправлення помилок.

В якості інформаційного критерію зазвичай виділяється кількість керівників бітів, які відправляються з блоком даних.

В даний час з розвитком технологій обміну інформацією відбувається зміщення акценту на важливість перерахованих вище критеріїв ефективності методів контролю помилок. Помітне зростання швидкості передавання інформації через мережі обумовлює збільшення важливого фактору оцінки обчислювальної складності процедур дешифрування, локалізації та виправлення помилок, що надалі розрахувати виконання згаданих процедур на швидкості передачі. Цей фактор знижує важливість кількості керуючих символів, які додатково відправляються в якості критерію ефективності.

Для виправлення множинних помилок поширеним є використання кодів корекції Ріда-Соломона, що представляють собою недвійкові циклічні коди. За допомогою цих кодів можна виправляти будь-яку кількість *h*, перекручених під час передавання символів. При цьому використовується 2**h* керуючих символів. При такому випадку потрібно враховувати, що ліміт на довжину блоку. Наприклад візьмемо *n* як кількість всіх символів блоку, а тоді маємо (*n*-4) інформаційні символи при умові якщо [15]:

$$n \leq 2^m - 1.$$

Простий поліном на полях Галуа ступеня *h* використовується для генерації коду Ріда-Соломона. Поліном для генерації *f(x)*, ступінь якого рівний загальній кількості бітів в керуючому коді 2**t*. Поліном *f(x)* у свою чергу утворюється як добуток $2*t$ складових:

$$f(x) = (x - \alpha) * (x - \alpha^2) * (x - \alpha^3) * \ldots * (x - \alpha^{2*t}).$$

Розглянемо такий приклад

$$f(x) = (x - \alpha) * (x - \alpha^2) * (x - \alpha^3) * (x - \alpha^4)$$
$$= x^4 + \alpha^3 * x^3 + \alpha^0 * x^2 + \alpha^1 * x + \alpha^3$$

З рівняння знаходимо, що коренями для *f(x)* буде $\alpha, \alpha^2, \alpha^3, \alpha^4$.

Тоді $f(x) = \alpha^4 + \alpha^3 * \alpha^3 + \alpha^0 * \alpha^2 + \alpha^1 * \alpha^1 + \alpha^3 = \alpha^4 + \alpha^6 + \alpha^2 + \alpha^2 + \alpha^3 = 0$.

При умові наявності інформаційного блоку, він буде братися як поліном *m(x)*. Щоб обрахувати керуючі біти він множиться на x^{n-k}, після чого береться залишок *p(x)* від ділення $x^{n-k} * m(x)$, помножений на породжуючий поліном *f(x)*. У результаті приймачеві відправляється блок (як поліном $U(x) = x^{n-k} * m(x) + p(x)$).

Легко помітити, що *U(x)* ділиться на *g(x)*. Це можна прослідкувати виходячи з того, що $U(x) = x^{n-k} * m(x) + p(x)$, що відповідає сумі полінома та залишку від ділення на *g(x)*. Якщо *U(x)* ділиться на *g(x)*, тоді *U(x)* ділиться на корені для *g(x)*, а це $\alpha, \alpha^2, \alpha^3, \alpha^4$. Як приклад $U(\alpha^3) = 0$.

Синдром помилки розраховується на стороні приймача. Якщо блок пройшов без помилок, то всі синдроми дорівнюють нулю.

Обчислювальна складність операції контролю виявлення помилок становить O(8**h***n***m*).

2**h* синдроми можна обчислювати паралельним чином, тому що їх можна розбити та виконувати обчислення окремо і незалежно. Проте розрахунок самого синдрому вимагає послідовного виконання, а це за собою веде до того, що тимчасова складність контролю помилок рівна O(4**n***m*).

Щоб виправити помилки передавання потрібно виконати наступну послідовність кроків:

1. рішення h символьних рівнянь для того, щоб визначити h складових $\delta_1, \delta_2, \dots, \delta_h$ полінома $\theta(X) = \delta_1 * X^{h-1} + \delta_2 * X^{h-2} + \cdots + \delta_{h-1} * X + \delta_h$ визначальника-локатора помилок. З цією метою формується обернена матриця. Вона отримується як результат частки ділення транспонованою матриці значень синдромів на визначник. Складові

полінома локатора помилок обчислюються шляхом множення отриманої зворотної матриці і вектора синдрому помилки. Сумарний час рішення системи рівнянь для отримання полінома помилки можна представити у вигляді $4 * h^4 * (t_m + t_{XOR})$.

2. знаходження положення перекручених під час передавання символів відбувається за допомогою визначення ненульових значень $\theta(\alpha), \theta(\alpha^2), \dots, \theta(\alpha^{n-1}), \theta(\alpha^n)$. Сумарний час, який потрібен для повної перевірки n підстановок можна представити у вигляді $n * h * (t_m + t_{XOR})$.
3. рішення системи рівнянь h символів для того, щоб отримати значення для h символів спотворення $e_1, e_2, \dots, e_h$. Сумарний час для рішення системи рівнянь для пошуку символів спотворення можна представити у вигляді $4 * h^2 * (t_m + t_{XOR})$.

У результаті сумарний час T_s для корекції спотворених символів можна представити формульно таким чином:

$$T_S = (n * h + 8 * h^2) * (t_m + t_{XOR}).$$

Таким чином обчислювальну складність процедур, пов’язаних з корекцією групи помилок можна визначити як O$(n * m * h^2)$.

Через те, що коди Ріда-Соломона використовують операції множення для полів Галуа, які не підтримуються процесорами, існують обчислювальні ресурси для управління і виправлення помилок.

Істотним мінусом коригувальних кодів, що були розглянуті вище, являється висока обчислювальна складність процесів визначення місця та виправлення спотворених символів. Це пов'язано з потребою вирішення систем нелінійних символьних рівнянь в арифметиці полів Галуа [4]. Як відомо, систему нелінійних булевих рівнянь можна вирішити тільки за допомогою перебору (брутфорсом). З огляду на архітектуру сучасних процесорів, яка не призначена для виконання операцій з полями Галуа, очевидним вирішенням проблеми виправлення помилок за допомогою кодів виправлення є спеціальне обладнання.

Обчислювальна складність імплементації виправлення залежить від кількості символів, які код може коригувати.

Erasure codes ефективніші з точки зору обчислювальних ресурсів і надмірності передачі даних [6, 7]. Істотна відмінність між кодами виправлення і erasure codes полягає в тому, що erasure codes не локалізують втрачені або пошкоджені блоки, ці операції здійснюються вбудованими інструментами. Більша частина erasure codes використовують лінійні коди (LT-кодування) в якості математичної основи, що забезпечує швидкість і ефективність обчислювальної імплементації операції відновлення даних. Найвідоміший тип erasure codes – Raptor [3]. Використовуючи його можна відновити будь-яке число втрачених блоків з n переданих блоків. Це досягається із застосуванням високої надмірності – понад 100%.

Існуючі erasure codes не враховують особливості викладеного раніше завдання відновлення відеоінформації під час трансляції відео. Наприклад, ліміт на число втрачених блоків і безперервність втрачених блоків даних (тобто ми можемо говорити про відновлення «групи» блоків).

1.4. Дослідження технологій відновлення даних при їх збереженні на віддалених носіях

Після того, як з’явилися клауд технології та поруч з цим нові вимоги пов’язані з передаванням даних через Інтернет, виникли слідом нові труднощі з відновлення даних. Клауд технології дозволяють багатьом користувачам комерційно на ремоуті використовувати ресурси процесора, ресурси пам'яті для зберігання даних і програмного забезпечення в якості платних послуг [1]. За рахунок динамічної зміни конфігурації, за допомогою якої можлива адаптація до змінного навантаження, здійснюється оптимізація процесів використання обчислювальних ресурсів і ресурсів пам'яті [3]. Це дозволяє клієнтам застосовувати потужні обчислювальні ресурси і можна сказати безлімітні обсяги пам'яті для вирішення задач на практиці. Згадуючи віддалене зберігання даних, потрібно сказати, що дані ділять та складають на окремих вузлах, призначених

для зберігання інформації. Всередині одного такого вузла поділ і поширення даних на носії, інформаційний доступ клієнтами, захист і зберігання в разі зникання доступу до одного або декількох вузлів впорядковано.

Схематичне зображення організації ремоут збереження даних клієнтів за допомогою клауд технологій можна побачити на рис. 1.1.

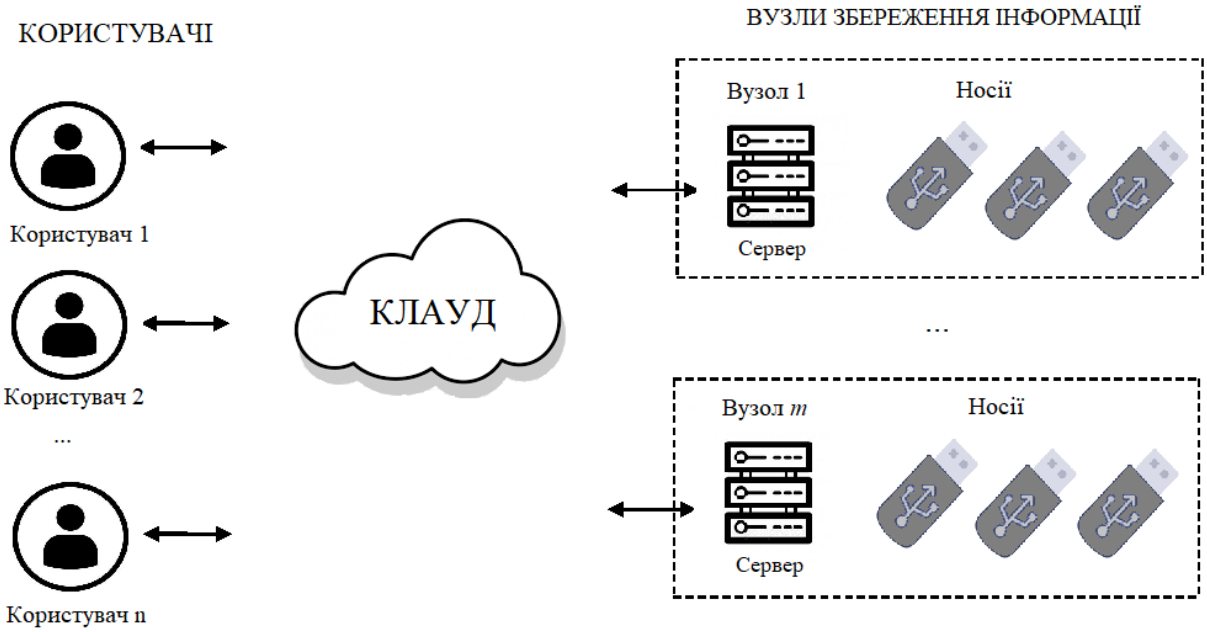

Рис. 1.1. Організації ремоут збереження даних клієнтів

Щоб отримати ефективне користування клауд технологіями для збереження ремоут, потрібно знайти рішення проблеми поділу ресурсів, інформаційної безпеки та секьюрності доступу до клауду.

Проблема надійності має важливе значення для технологій, що дозволяють надавати користувачам ремоут ресурси для зберігання даних. Іншими словами, при розгляді ремоут зберіганні даних клієнта, ми повинні враховувати такий факт: ремоут носії непередбачувані і доступ до них можливий третім особам. Ці фактори призводять до ризику пошкодження чи втрати даних, що зберігається в клауді. Існують приклади, де клієнти не могли доступитися до збережених на сервері даних довгий час.

Сучасні локальні мережі стали настільки великими, що з технічних і топологічних характеристик їх можна порівняти з глобальними мережами. Як

бачите, завдання зберігання інформації та надання до неї доступу в глобальних і локальних мережах вимагають спільного рішення. У той же час за вмістом технології організації розподілених мереж зберігання даних мають різні рішення. В даний час найбільш перспективними технологіями для створення ремоут сховища є

- SAN (Network Area Storage / Мережеві Пристрої Зберігання)
- DAS (Storage Attached Direct / Пристрої Прямого Підключення до Серверів)
- NAS (Storage Attached Network / Пристрої, які Підключаються через Internet).

У той же час NAS все більше використовується в глобальних мережах, а SAN – в локальних [8].

Щоб забезпечити безперервний якісний доступ кожному клієнтові до інформації, що зберігається в віддалених репозиторіях, найчастіше виконують резервну копію цієї інформації.

Серед критеріїв ефективності резервного копіювання даних можна виділити наступні:

- обчислювальна складність операції відновлення втраченої інформації, використовуючи резервні носії;
- число носіїв E_m – відповідає кількості носіїв, на яких дані можуть бути відновлені при необхідності;
- відношення між числом носіїв, на яких збережені резервні копії E_r і числом носіїв, на яких дані можуть бути відновлені: $\alpha = \frac{E_m}{E_r}$.

Дослідження літературних джерел говорить про те, що втрата даних, збережених на ремоуті, часто спричинена техногенними факторами, поломка або профілактичні роботи з репозиторієм. Іншими словами, коли доступ до інформації відсутній чи дані були ціле направлено атаковано та пошкоджено.

Під час випадків, коли інформація губиться на репозиторіях за їх збою, через помилке або зловмисне стирання інформації, число носіїв q залежить від часу t, що пройшов з моменту останнього доступу до інформації.

Нехай клієнт використовує N носіїв для зберігання даних, вірогідність втрачання інформації від часу $P(t)$ слідує експоненціальному закону. З цього вірогідність ситуації, коли до k носіїв втратиться доступ, може бути визначена за формулою:

$$P(t) = C_N^k * (1 - e^{-\gamma * t})^k * e^{-(N-k)*\gamma*t}, \quad (1.1)$$

де γ – інтенсивність втрачання даних з носія через збій або видалення інформації. Зробивши аналіз випадків відмови носія та керуючись формулою (1.1), можна дійти до висновку, що вірогідність втрачання доступу до одного носія досить низька, а вірогідність випадків відмови у 2-3 рази нижче. Звідси виходить, що при випадку, коли час доступу до носія становить не більше року, число носіїв, доступ до яких втрачається за причин збою чи стирання інформації, не перевищує трьох. Більшість реальних систем резервного копіювання інформації мають 3 носії, доступ до яких може бути втрачено.

На даний момент найбільш популярною являється технологія RAID. Він заснований на розділі зберігання інформації та виконанні резервного копіювання.

В порівнянні з однодискової системою зберігання даниї технологія RAID підвищує надійність зберігання даних. Для того, щоб не втратити інформації, що зберігається на диску за причини збою, ви можете відновити відсутні дані з інших дисків. RAID збільшує продуктивність введення-виведення і підвищує надійність підсистеми збереження.

Поділ дисків є важливою складовою технології RAID, при якому передача даних переноситься на мітку диска. Розподіл дисків для перенесення даних має своєю суттю розбивку дискового простору на смуги, розмір яких може бути від 8 КБ до 1024 КБ. Вони почергово змінюються на регулярній основі послідовно.

Спільне місце, призначене для збереження містить смуги з усіх пристроїв. Рекомендується, щоб смуги мали однаковий розмір у групах RAID-дисків [15].

Розглянемо приклад, коли 4 диски в системі, де відбувається поділ дисків, запис сегменту 1 виконується на диск 1, сегменту 2 – на диск 2 і відповідно далі. Поділ дисків збільшує продуктивність за причини, що є доступ одразу до декількох дисків, проте почерговість дисків не впливає на надмірність інформації.

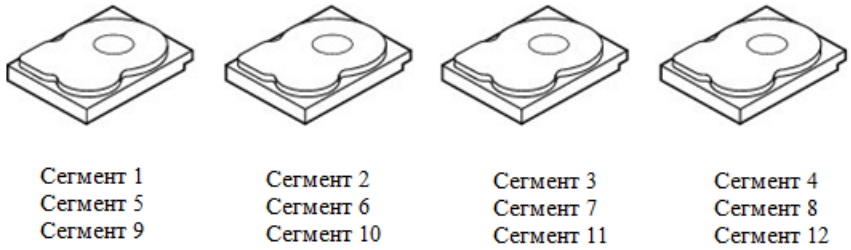

Рис. 1.2. Поділ дисків (RAID 0)

Ширина смуги – число дисків в групі дисків у місці поділу. Як приклад, група з чотирьох дисків з поділом дисків має ширину смуги рівну чотирьом.

Розмір смуги – довжина сегментів інформації, що йдуть почергово та запис яких RAID-контролер виконує на декілька дисків, незалежно від парності дисків. Як приклад, смуга містить 64 КБ дискового простору і 16 КБ даних, розташованих на кожному диску смуги. При такому розкладі маємо смугу 64 КБ і смугу в 16 КБ. Розмір смуги – це частина смуги, яка розташована на тому ж диску.

Віддзеркалення дисків вважається ще однією суттєвою складовою технології RAID. Відбувається запис інформації на один диск і одночасно на інший. Головна перевага дзеркального копіювання дисків в тому, що воно забезпечує 100% надмірність даних. Через те, що дані одного диска повністю записано на другий диск, вони не втратяться за випадку виходу з ладу одного диску. Плюс до цього, оскільки ці два диски мають одні й ті ж постійні дані, один з дисків може бути як оперативний диск. Якщо один диск виходить з ладу,

другий диск можна використовувати для завантаження системи і процесу відновлення іншого пошкодженого диска.

Віддзеркалення дисків як було вище згадано надає 100% надмірність резервування, але є затратним, оскільки кожен диск системи має дублюватися.

Гаряче резервне копіювання – додатковий невикористаний диск, який є гарячим резервом та є частиною дискової підсистеми. Зазвичай він знаходиться в режимі очікування і готовий до роботи в разі виходу диска з ладу. У ситуації, коли диск перебуває у використанні в віртуальному диску RAID і неочікувано перестає працювати, «гарячий» резерв автоматично заміняє його, і дані на цьому диску відновлюються в «гарячому» резерві без виключення системи чи втручання клієнта. Гарячий резерв може застосовуватися для RAID 1, 5, 6, 10, 50 і 60 рівнів.

Гаряче резервування розділяють на такі види:

- глобальне гаряче резервування;
- виділений оперативний резерв

Глобальне гаряче резервування – випадок, коли глобальний резервний диск може застосуватися, щоб заміни несправного диска в резервної групі дисків, якщо його ємність дорівнює або перевищує обсяг відмовившого диска. Глобальний операційний резерв, певний на будь-якому каналі, має бути доступним при заміні диску на обох каналах при відмові.

Виділений оперативний резерв – виділений оперативний резерв можна застосовувати, щоб заміняти несправний диска тільки у вибраних групах дисків. Один або кілька дисків можуть бути включені в запасний диск. Для запису в разі виходу з ладу вибирається найбільш підходящий диск. Виділений оперативний резерв застосовується до одного з глобальних оперативних резервів [13].

Контролер RAID використовує гарячі резерви для автоматичного і прозорого відновлення, коли відмовили диски, за допомогою визначені клієнтом масштабів відновлення. За умови наявності гарячого резервного копіювання, запуск відновлення може бути автоматичним в разі виходу з ладу одного з

дисків. Інакше, несправний диск необхідно замінити новим, щоб можна було відновити дані на ньому.

Несправний диск видаляється з віртуального диска і позначається як готовий до видалення, коли починається відновлення з гарячою резервної копії. У ситуації, коли система виходить з ладу у процесі відновлення, контролер RAID автоматично перезапускає відновлення після перезавантаження системи.

Запуск автоматичного відновлення диска не відбувається за ситуації, коли жорсткий диск був замінений на час міграції рівня RAID. Реструктуризація має запускатися вручну після розширення або після завершення процедури міграції. Міграція рівня RAID – це зміна віртуального диска з одного рівня RAID на інший.

Здатність системи проходити через перебої без небезпеки для цілісності даних і можливостей обробки визначає відмовостійкість. Контролер RAID не дозволить отримати доступ до груп дисків збільшеного розміру на рівнях RAID 1, 5, 6, 10, 50 і 60. Систему може залишатися працездатною при відмові диска в групі дисків, проте продуктивність може бути знижена до визначеного рівня.

- Група дисків RAID 1 складається з двох дисків і може витримати відмову одного диска.
- Група дисків RAID 5 може витримати відмову одного диска в кожній групі дисків RAID 5.
- Група дисків RAID 6 витримує відмову відразу двох дисків.
- Кожен віртуальний диск RAID 10 може витримувати відмову декількох дисків до моменту, коли кожен збій не відбудеться в окремій групі дисків.
- Віртуальний диск RAID 50 може витримати відмову двох дисків до моменту, коли кожна відмова не відбудеться в окремій групі дисків.
- Дискова група RAID 60 може витримати до двох відмов дисків в кожної дискової групі.

Часто відмовостійкість зв'язана з системою доступності, бо дає можливість мати доступ до системи у ситуаціях збою. Тім не в останню чергу, важливо, щоб система була доступна протягом усунення проблеми.

Автоматична реорганізація надає можливість виконати заміну диску, що вийшов з ладу, а дані автоматично відновлюються шляхом заміни гарячого диска. RAID групи продовжують опрацьовувати запити під час відновлення.

Для будь-якої відмовостійкої конфігурації ви можете створити додатковий захист від втрати даних, призначивши фізичний диск в якості оперативного резервного копіювання (або «гарячого резервування»). Диски резервного копіювання не мають інформації та їх необхідно тримати в тому ж масиві, що і логічний диск, якому вони призначені. Логічному диску може бути призначено кілька запасних фізичних дисків, обмеження накладається тільки наявністю невикористовуваних дисків в масиві.

При виході з ладу фізичного диска з масиву контролером здійснюється автоматично відновлення даних з пошкодженого диска в оперативний резерв. Відновлення системи проходить швидко при повному захисті даних RAID. При цьому низька вірогідність того, що інший диск масиву вийде з ладу під час перезапису даних, логічний диск може вийти з ладу, в залежності від використовуваної конфігурації RAID [12].

Окрім резервного копіювання, поширене використовуваного в розподілених системах збереження інформації, часто використовуються коригувальні коди.

Найбільш широкого практичного застосування отримали технології відновлення інформації, що ґрунтуються на кодах корекції і відновлюючих кодах (erasure codes). У випадку, коли відновлюється інформація з носія, доступ до якого втрачено, зазвичай немає необхідності їх локалізувати. Класичні коди корекції, наприклад коди Хемінгуея, Ріда-Соломона, спрямовані на почергове здійснення двох операцій: локалізацію спотвореної частини інформації і процес її виправлення. Виходячи з цього за застосування кодів корекції, згаданих вище, для відновлення інформації з носія, до якого втратили доступ, необхідно їх

модифікувати. Наприклад для відновлення інформації з носіїв RAID-6 використовують модифіковані коди Каучі–Ріда-Соломона.

Для цього більш ефективно застосовувати спеціальні відновлюючі коди. Більшість цих кодів засновані на лінійних перетвореннях, що викликає швидке збільшення числа носіїв резервних копій зі збільшенням кількості носіїв, до яких втрачається доступ [4].

Спільна особливість поширених технологій відновлення даних з носіїв, до яких загублений доступ, - це їх реалізація на окремому вузлі збереження. Тому при втраті доступу до вузла через його тимчасовий збій, перевантаження, вірусні атаки, відключення від мережі, техногенні чи стихійні лиха відновлення за допомогою відомих механізми відновлення даних або доступ до них для конкретного клієнта не працює.

З цього можна зробити висновок, що наявні методи відновлення доступу до інформації в своїх ремоут системах збереження даних не гарантують рішення цієї проблеми в ситуації втрати доступу до вузла збереження даних.

Також доведено, що наявні методи резервного копіювання інформації мають високий рівень надмірності і кількість носіїв резервних копій можна зменшити без втрати можливості відновлення даних.

Висновки до розділу 1

За результатами першого розділу дипломного проекту був виконаний аналіз поточного розвитку інформаційних технологій та технологій передачі інформації через глобальні мережі, вивчені тенденції розвитку механізмів забезпечення надійності передачі і віддаленого зберігання даних. Розглянуто існуючі методи виправлення помилок при обміні даними. На підставі цього можна зробити наступні висновки:

1. Зі швидким розвитком глобальних мереж відбувається динамічне зростання швидкості передачі даних і зростання частки ефірних каналів, передача по яких може викликати втрату блоків даних, викликану зовнішніми перешкодами. Зростання швидкості передачі інформації збільшує ймовірність помилок і втрати даних. В даний час існує тенденція до збільшення швидкості передачі даних, в той час як нові інструменти і методи розробляються для відновлення втрачених або спотворених блоків даних при передачі в глобальних мережах.
2. Описано та обґрунтовано критерії ефективності механізмів, що забезпечують надійність при передачі даних через глобальні мережі. Огляд літератури показав, що критерій часу передачі даних набуває все більшого значення, що особливо важливо для комп'ютерних систем реального часу. Був виконаний аналіз критерій кількості керуючих символів, що передаються з даними і використовуваних при виправленні помилок, що виникають при передаванні при прискоренні обміну інформацією. Він показав, що згаданий критерій поступається за важливістю. В результаті з'являються нові інструменти і методи відновлення даних, які підтримують підвищену продуктивність за рахунок зростання надлишкової додаткової інформації.

РОЗДІЛ 2

РОЗРОБКА МЕТОДУ ПРІОРИТЕТНОГО ВІДНОВЛЕННЯ БЛОКІВ ДАНИХ

2.1. Теоретичне обґрунтування

Класичний варіант погляду на проблему включає тільки вирішення задачі корекції даних. Коди корекції створені та використовуються для рішення цієї проблеми. Щоб виправити помилки, потрібно виконати три кроки:

- знайти випадок, при якому виникла помилка;
- локалізувати помилку;
- виправити помилку [2].

Однак в сучасних розробках прослідковується тенденція зсуву уваги застосування коригувальних кодів на застосування кодів відновлення на рівні блоків. Це пов'язано з тим, що раніше відновлювали втрачену інформацію на рівні бітів або символів, тепер будемо розглядати відновлення на рівні блоку. Час, за який дані можливо відновити, є найбільш важливим фактором, який беруть до уваги щодо відновлення. На другому місці за важливістю ставлять такий фактор, як клас виправлення помилок. Після них ставлять такий фактор як обсяг додаткової пам’яті, яку необхідно залучити. Це спричинено ситуацією поступового зниження вартості пам'яті.

За термінологією у спеціалізованих літературних джерелах замість кодів корекції використовують термін кодів відновлення – erasure codes.

Переважаючим фактором виступає швидкість виправлення помилок і відновлення даних, що вплинуло на рішення використання лінійних кодів – кодів, сформованих шляхом лінійних перетворень над інформаційними блоками [1].

Відбувається передача повідомлення з інформацією поділене на n блоків, які посилаються альтернативними шляхами. Блок по номеру i та $i \in \{1, \dots, n\}$

вміщає групу слів, що послідовно йдуть, $w_{i1}, w_{i2}, \dots, w_{im}$. У разі втрати при передаванні або недостовірність даних, збережених в блоках по номерах p та q, де $p, q \in \{1, \dots, n\}$, потрібно виконати відновлення значень, що містяться у словах: $w_{p1}, w_{p2}, \dots, w_{pm}$та $w_{q1}, w_{q2}, \dots, w_{qm}$ відповідно до даних, збережених у $n-2$ основних і k резервних блоках. Для забезпечення легкого процесу відновлення загублених та недостовірних даних пропонується виконати формування першого резервного, виконавши обчислення суми за модулем 2 всіх відповідних слів, що передаються в основних блоках:

$$\forall j = 1, \dots, n: S_{1,j} = \oplus_{i=1}^{n} w_{ji}. \quad (2.1)$$

При розгляді, щоб відновити p-тий і q-тий основні блоки слів : $w_{p1}, w_{p2}, \dots, w_{pm}$та $w_{q1}, w_{q2}, \dots, w_{qm}$, можемо обчислити їх значення вирішивши систему з двох булевих рівняння:

$$\forall j = \{1, \dots, n\}: \begin{cases} w_{pj} \oplus w_{qj} = r_{1j} \\ w_{pj} = r_{2j} \end{cases} \text{ чи } \begin{cases} w_{pj} \oplus w_{qj} = r_{1j} \\ w_{qj} = r_{3j} \end{cases} \quad (2.2)$$

Символи «$\oplus$» позначають операцію порозрядного додавання по модулю 2. У системі (2.2) 1-ше рівняння виходить перетворенням з рівняння (2.1), відповідно, щоб одержати 1-ше рівняння застосовують дані, що зберігаються в 1-ому блоці резервного копіювання. Через те, що на початку невідомо значення p та q, одержати 1-ше рівняння з системи (2.2) набагато складніше. Видно, що 2-ге рівняння з системи (2.2) можемо одержати з системи лінійних булевих рівнянь, яка при $\forall p$ та $\forall q$ вміщає рівняння, в якому член включає лише w_{p1}чи w_{q1}. При n кратному 2 будемо мати прикладом таку систему:

$$\forall j \in \{1, \dots, n\}: \begin{cases} w_{1j} \oplus w_{2j} \oplus \dots \oplus w_{\frac{m}{2},j} = y_j \\ w_{1j} \oplus w_{2j} \oplus \dots \oplus w_{\frac{m}{4},j} \oplus w_{\frac{m}{2}+1,j} \oplus \dots \oplus w_{\frac{3m}{4},j} = y_2 \\ \dots \\ w_{1j} \oplus w_{3j} \oplus w_{5j} \oplus \dots \oplus w_{m-1,j} = y_{\log_2 n,j} \end{cases}. \quad (2.3)$$

У загальному випадку кількість рівнянь в системі виду (2.3) за умови довільного значення n рівне найближчому цілому числу, яке дорівнює або більше за $\log_2 n$: $\lceil \log_2 n \rceil$. Щоб відновити два блоки (основний або резервний), вам необхідно зарезервувати суму в першому резервному блоці по модулю 2 слів

з тим же ім'ям основних блоків (2.1), суми, належні системі (2.3) в $\lceil \log_2 n \rceil$ резервних блоках. і скопіюйте останній основний блок в інший резервний блок.

Коли слово з двох блоків, що втратили чи доставили недостовірними при передаванні даних, відновлюється незалежно, кількість k_p резервних блоків так званих носіїв, тоді маємо $k_p = 2 + \lceil \log_2 n \rceil$.

Організовується резервне копіювання даних так, що вона не вимагає витрат з точки зору обчислювальної складності та надає гарантію відновлення загублених чи недостовірних даних. Найбільш важливим є той факт, що відновлення може бути виконано зі швидкістю передачі даних. Проте недолік в тому, що такій організації притаманна висока надмірність копіювання.

Досліджуючи дане питання розглянули різні комбінації передавання і втрати основних і резервних блоків.

У випадку втрати 2-ох блоків:

1. Втрачено два основні блоки.

   Блоки, що були втрачені при передачі, можна локалізувати кількістю способів рівних $C_7^2 = \frac{7*6}{2} = 21$. За 3-ьох локалізацій стовпчики однакові 1000100...0010001. З цього кількість випадків, коли можна відновити дані дорівнює 21 – 3 = 18, у той час, коли відновити не можна – 3-ьом.

2. Втрачено один основний та один резервний блоки.

   За такого випадку відновити основний блок при втраті можна завжди.

Звідси всіх можливих випадків втрачання попарно блоків з 10-и – 7-и основних та 3-х резервних буде $C_{10}^2 = \frac{10*9}{2} = 45$. Та існує всього 3 випадки, за яких відновлення виконати не можна. Таким чином, вірогідність корекції $p_2 = \frac{45-3}{45} = \frac{42}{45} = 0{,}933$.

У випадку втрати 3-ох блоків:

1. Втрачено три основні блоки.

   Виправлення неможливе при наявності у двох втрачених стовпців однакових символів, що становить три випадки. Проте кожний випадок

включає п’ять випадків знаходження третього втраченого блоку. Таким чином, сумарне число випадків, за яких виправити не можна дорівнює 15 випадків (3 * 5 = 15).

2. Втрачено два основні та один резервний блоки.

   Зі стовідсотковою ймовірністю неможливо виконати відновлення за умови, що у двох втрачених блоків є зв’язок з одними і тими ж стовпцями, що становить три випадки. Сумарне число випадків враховуючи встановлення втраченого резервного блоку становить 3 * 3 = 9 випадків.

   Також можливо, що буде випадок, коли один втрачений блок – це четвертий блок, що співвіднесений зі стовпцем 111, а другий – інший довільний, що відповідає шістьом випадкам. При цьому, за умови, що ви втратите резервний блок, відповідний рядку, значення стовпчика якого мають різні значення, блоки не можуть бути відновлені. З цього, сумарне число випадків, за яких відновити втрачені блоки не можна дорівнює 9 + 6 = 15.

3. Втрачено два резервні і один основний блоки.

   Основний блок не можна відновити за випадку, коли це не четвертий і втратилися два резервних блоки, що співвідносяться з одиницями в стовпці. Тому видно, що число таких випадків 6.

Зробимо висновок: із сумарного числа, шо становить $C_{10}^{3} = \frac{10*9*8}{3*2} = 120$ різних випадків, коли ми хочемо локалізувати 3 втрачені блоки, маючи 7 основні та 3 резервні блоки, стикаємося з 33 випадками, за яких відновити втрачені основні блоки не зможемо (15 + 12 + 6 = 33). З цього випливає, що вірогідність при передачі відновити втрачені блоки $p = \frac{120-33}{120} = 0{,}725$.

Також було досліджено модель, яка містить різні стовпці.

У випадку втрати 2-ох блоків:

1. Втрачено два основні блоки.

   У такій ситуації відновлення двох основних блоків можливе постійно.

2. Втрачено один основний та один резервний блоки.

У такому випадку відновлення основного блоку можливе завжди за умови, що число одиниць відповідного стовпця матриці векторів λ більше 1. У випадку, коли число одиниць рівне 1 і втрата прийшлася на відповідний резервний блок, то відновити втрачений блок не вдасться. Можна стикнутися з трьома такими випадками.

З цього слідує, що при $C_{10}^{2} = 45$ випадках, коли втратили пару блоків з десяти блоків, існує тільки три випадки, коли не можна їх відновити. Тоді вірогідність, що відновлення вдасться $p_2 = \frac{45-3}{45} = \frac{42}{45} = 0{,}933$.

У випадку втрати 3-ох блоків:

1. Втрачено три основні блоки.

У випадку, коли втратили три основні блоки, відповідні стовпці різні. Тоді не можна утворити ортогональну систему за умови, коли стовпець дорівнює сумі по модулю 2 двох інших стовпців. Зрозуміло, що попарно стовпці утворюють ці суми, їх число становить 7, тому що представлення за допомогою суми по модулю два двох інших стовпців для кожного стовпця може бути виконано трьома способами.

2. Втрачено два основні та один резервний блоки.

Коли один резервний блок втрачається в матриці θ, формуються один стовпець та три пари ідентичних стовпців. За випадку, коли один з основних втрачених блоків пов'язаний зі стовпчиком під номером 0, його неможливо відновити. Одночасно з тим другий втрачений блок можна відновити. Загалом, коли втрачений блок корелює з нульовими стовпцями, існує 6 можливих випадків локалізувати другий втрачений основний блок. Плюс за ситуації, коли нульові стовпці матриці θ пов'язані з втраченим блоком, є 3 випадки, коли пара втрачених основних блоків корелюється з одними і тими ж стовпцями.

Тоді, для кожної з трьох локалізацій втраченого резервного блока є 6 + 3 = 9 випадків локалізувати основні втрачені блоки, за яких їх неможливо відновити. Тому сумарне число випадків рівне 3 * 9 = 27.

3. Втрачено два резервні і один основний блоки.

За випадків, коли локалізуємо пари резервних блоків, що втратили, три стовпці матриці λ набувають нульового значення, а тому можна виділити 3 випадки, коли локалізувати втрачений основний блок не можна. Загалом є 3 * 3 = 9 випадки, коли локалізувати за ситуації втрати одного основного і двох резервних блоків, за якої втратили основний блок і його відновити неможливо.

Зробимо висновок: коли три блоки губляться із сумарного числа випадків $C_{10}^{3} = 120$, є 7 + 27 + 9 = 43 випадки, за яких основні блоки, що загубилися неможливо відновити. Отже, вірогідність того, що втрачені у процесі передачі блоки будуть відновлені $p = \frac{120-43}{120} = \frac{77}{120} = 0{,}642$.

Щоб досягнути мету у роботі запропонували метод формування та застосування блоків резервних копій для того, щоб відновлювати втрачену інформацію. Блоки резервного копіювання – це лінійна комбінація основних блоків.

Для зручності ввели наступні позначки:

n – кількість основних блоків з інформацією;

m – кількість блоків з резервними копіями;

u – кількість блоків, що втравтили;

P_i – основний блок з інформацією, де $i \in \{1,2,\dots,n\}$;

R_j – блок з резервною копією, де $j \in \{1,2,\dots,m\}$;

B_j – бінарний вектор, такий, що $B_j = \{B_{j1}, B_{j2}, \dots, B_{jm}\}$, де $B_{ji} \in \{0,1\}$;

R_j – сума по модулю 2 основних блоків:

$$R_j = \oplus_{i=1}^{m} B_{ij} * P_j.$$

Відправляється $n + m$ блоків по мережі. Зрозуміло, що якщо u блоків втратили, пошкодили чи запізнилися під час передачі, одержувач отримає $m + n$

– u блоків. З цього випливає умови, що потрібні, щоб мати можливість відновити u блоків:

1. число блоків, що втратили, повинне становити менше число або рівне число числу блоків з резервними копіями $u \leq m$;
2. блоки, одержані під час передавання можна було використати для формування ортогональної системи з n рівняннями.

Мірою ефективності відновлення блоків інформації, втрачених під час передачі, є вірогідність їх відновити.

За випадку, коли втратили один основний блок P_i, ортогональна система має утворитися з векторів $V_1, V_2, \dots, V_{i-1}, \dots, V_{i+1}, \dots, V_n$ та одного із додаткових векторів $B_1, B_2, \dots, B_m$. А тому повинен бути такий вектор B_j, що його i-а складова B_{ji} дорівнювала б одиниці (B_{ji} = 1). З цього випливає, що довільна сума підмножини векторів $V_1, V_2, \dots, V_{i-1}, \dots, V_{i+1}, \dots, V_n$ не рівні вектору B_j, а тому вони формують ортогональну систему лінійних рівнянь. Рішення цієї системи дасть змогу отримати вектори V_i, що втратили, у форматі лінійної комбінації векторів $V_1, V_2, \dots, V_{i-1}, \dots, V_{i+1}, \dots, V_n, B_j$:

$$a_j \in \{0,1\}:$$

$$V_j = a_1 * v_1 \oplus a_2 * v_2 \oplus \dots \oplus a_{i-1} * v_{i-1} \oplus a_i * v_i \oplus a_{i+1} * v_{i+1} \oplus a_n * v_n.$$

Тоді відновити блок, що втратили, наступним чином:

$$P_j = a_1 * P_1 \oplus a_2 * P_2 \oplus \dots \oplus a_{i-1} * P_{i-1} \oplus a_i * P_i \oplus P_{i+1} * v_{i+1} \oplus a_n * P_n.$$

Щоб стовідсотково відновити 1 блок інформації потрібно, щоб вектори $B_1, B_2, \dots, B_m$ не містили б відповідних складових, що одночасно були б нульовими. Так що досить 1-ого додаткового, тобто m = 1, складові якого рівні одному. З цього i-ий блок можна відновити користуючись формулою:

$$P_i = P_1 \oplus P_2 \oplus \dots \oplus P_{i-1} \oplus D \oplus P_{i+1} \oplus \dots \oplus P_n.$$

Щоб відновити дані 2-ох основних блоків, що втратили, що мають номери l і k, де $l, k \in \{1,2, \dots, n\}$, необхідно виконання умов:

$$\begin{cases} \gamma_{xl} \oplus \gamma_{yl} = 1 \\ \gamma_{xk} \oplus \gamma_{yk} = 1 \end{cases}$$

Зрозуміло, виконання останньої умови можливе тільки, якщо вектор λ_x є інверсією вектора λ_y. Проте в такій ситуації вектори не являються лінійно незалежними, а це означає, що якщо втратимо два блоки і використаємо два додаткові блоки, то сформувати ортогональну систему двох рівнянь нам не вдасться. Тому, ми довели, що не можна гарантувати виправлення двох блоків, якщо застосовуємо тільки пару додаткових блоків.

Взагалі кажучи, є k n-бітових кодів, при цьому вірогідність того, що вони містять n кодів, які можуть сформувати ортогональну систему, можна визначити наступним чином: вибирають довільний 1-ий код k_1, потім 2-ий код k_2 може набувати 2^n варіантів значень з умовою, що коди k_1 і k_2 не знаходяться в однаковій ортогональній системі. Іншими словами, код k_2 не збігається з k_1 чи код k_2 не зворотній коду k_1.

Зрозуміло, що вірогідність того, що код k_2 не міститься в ортогональній системі буде $1\text{-}2^{-(n-1)}$. Таким же чином 3-ій код k_3 також може набувати 2^n варіантів значень і не може бути частиною ортогональної системи в один і той же час, що коди k_1 і k_2, з умовою, що код k_3 не збігається з кодами k_1 і k_2 і не зворотній кодам k_1 і k_2.

Тоді $2 * (2^2 - 1)$ віднімається від усіх можливих значень 2^n, і вірогідність того, що 3-ій код k_3 може увійти в ортогональну систему поряд з кодами k_1 та k_2, представляється відповідно $1 - 3 * 2^{n-1}$.

Подібно до попереднього, виконуючи подальші обчислення, ми можемо зробити висновок, що вірогідність P_n – ситуація, коли n рандомних n-бітових кодів формують ортогональну систему з лінійних векторів, обчислюючи множенням вірогідностей, коли кожний код не можна представити лінійною комбінацією до цього обраних кодів:

$$P_n = \prod_{i=1}^{n-1}(1 - \frac{2^i - 1}{2^{n-1}}).$$

Таким чином, доведена ефективність використання лінійних кодів при рішенні завдання резервного копіювання інформації і відновлення втрачених

блоків даних, що відправлені через глобальні мережі або зберігаються у репозиторіях.

Важливим критерієм є те, що для довільного поділу вірогідностей втрачання певного числа блоків даних можливе утворення набору лінійних функцій для генерації резервних лінійних кодів.

Одним із чинників, що потенційно може впливати на час відновлення втраченого блоку є його довжина – r. Якщо фіксовано загальний об'єм V інформації, яка віддалено зберігається, то число n блоків при фіксованій їх довжині r визначається як $n = V/r$.

Час t_T транспортування блоку даних по мережі пропорційний довжині блоку r : $t_T = w \cdot r$ (w-швидкість передачі). Аналогічно, час t_{XOR} виконання базової обчислювальної операції реконструювання даних – логічного додавання також пропорційний довжині блоку r: $t_{XOR} = r \cdot \tau_{XOR}$ (τ_{XOR} – час виконання команди логічного додавання).

Аналіз публікацій [34] присвячених статистиці втрат інформаційних блоків при їх зберіганні на носіях показує, що в переважній більшості ймовірність втрати блоку не залежить від його довжини. Це зумовлено тим, що в процесі форматування носія здійснюється тестування кожного з секторів і запис інформації на пошкоджені сектори не виконується; на кожному секторі є вбудовані засоби локалізації та виправлення двох байтів, спотворених в процесі зберігання. Іншими словами, поступові відмови запам'ятовуючих елементів робочої поверхні носія складають в сучасних умовах відносно невеликий відсоток втрат доступу до інформаційних блоків. З цього можна зробити висновок про те, що за фіксований час, основним чинником, який визначає кількість втрачених блоків є їх загальна кількість n.

Якщо виключити інформаційні втрати, пов'язані з порушенням доступу до сховища в цілому, зумовлені цілеспрямованими хакерськими атаками та військовими і природними катаклізмами, то згідно з [35] процес втрат блоків даних описується біноміальною моделлю. Це означає, що за фіксований час

кількість втрачених блоків даних може бути обчислена за формулою Бернуллі, а середня їх середнє число становить $q \cdot n$, де q - ймовірність втрати блоку за вказаний вище час. З урахуванням наведеного, загальний час T_s відновлення втрачених інформаційних блоків оцінюється за наступною формулою:

$$T_s = T_m \cdot q \cdot n = q \cdot \frac{V}{r} \cdot (\varepsilon \cdot w \cdot r + r \cdot \tau_{XOR}) = q \cdot V \cdot (\varepsilon \cdot w + \tau_{XOR}) \quad . \qquad (2.4)$$

Аналіз формули (2.4) свідчить про те, що час відновлення інформаційних блоків при їх віддаленому зберіганні не залежить від їх розміру.

При відновленні інформаційних пакетів, втрачених, чи затриманих понад критичний час, в процесі передачі по глобальній мережі, судячи з публікацій [36] ситуація інша: ймовірність виникнення помилок передачі залежить від довжини пакету. Відповідно, ймовірність виникнення значної затримки доставки пакету, зумовленою його повторною передачею в результаті появи великої кількості помилок, що не можуть бути виправлені вбудованими засобами пакету, також залежить від його довжини. Якщо позначити через g ймовірність втрати одиниці інформації (байт) пакету, то ймовірність ρ_0, що при передачі не втрачається інформація становить: $\rho_0 = (1\text{-}g)^r$. Для подальшого аналізу більш зручним є представлення ρ_0 у вигляді суми:

$$\rho_0 = (1-g)^r = 1 - r \cdot g + \frac{r^2}{2} \cdot g^2 - \frac{r^3}{3!} \cdot g^3 + ... + g^r \quad . \qquad (2.5)$$

Враховуючи, що на практиці значення g становлять близько $10^{-4}…10^{-5}$ [37], то адитивні компоненти в формулі (2.5) швидко зменшуються, так, кожний наступний із них на порядки менший за попередній. Тому, з достатньою для задач аналізу можна враховувати лише ті компоненти ступінь при значенні g яких не більше h_1 – кількості помилок, які виправляються вбудованими корегуючи ми кодами на рівні пакетів. Тобто, формула (2.5) з урахуванням сказано може бути трансформована до вигляду:

$$\rho_0 \approx 1 - r \cdot g + \frac{r^2}{2} \cdot g^2 + \frac{r^3}{3!} \cdot g^3 - ...(-1)^{h-1} \cdot \frac{r^{h-1}}{(r-1)!} \cdot g^h .$$

Ймовірність ρ_1, що втрачається один байт пакету визначається наступним виразом:

$$\rho_1 = r \cdot (1-g)^{r-1} \cdot g = r \cdot g \cdot (1 - (r-1) \cdot g + C_{r-1}^2 \cdot g^2 - C_{r-1}^3 \cdot g^3 + ... + g^{r-1}) .$$

З огляду на наведені вище резони щодо можливостей спрощення формул з урахуванням реальних значень параметрів, що в них входять, значення ρ_1 може бути з допустимими в рамках поточного аналізу похибками представлено в наступному вигляді:

$$\rho_1 \approx r \cdot g - r^2 \cdot g^2 + \frac{r^3}{2} \cdot g^3 - ...(-1)^{h-1} \cdot r^{h-1} \cdot g^h .$$

Аналогічним чином можна показати, що ймовірність ρ_2 того, що втрачаються два байти пакету з достатньою для аналізу точністю, може бути представлена у вигляді:

$$\rho_2 \approx \frac{r^2}{2} \cdot g^2 - \frac{r^3}{2} \cdot g^3 + \frac{r^4}{4} \cdot g^4 - .... + (-1)^h \cdot r^h \cdot g^h .$$

Пакет повторно передається, якщо помилки сталися більш як в h байтах. Ймовірність P_h того, що кратність помилки перевищує h – кількість помилок, що виправляються вбудованими в структуру пакету корегуючи ми кодами, обчислюється у вигляді:

$$P_h = C_r^{h+1} \cdot (1-g)^{r-h-1} \cdot g^{h+1} + C_r^{h+2} \cdot (1-g)^{r-h-2} \cdot g^{h+2} + ... \quad (2.6)$$

Як зазначалося вище, для значень ймовірності g, що мають місце на практиці, величина кожної з компонент суми (2.6) визначається степенем g, так, що g^{h+1} на порядки більше за g^{h+2}. З урахуванням цього факту, ймовірність P_h, з достатньою для задач поточного аналізу точністю, можна наближено представити у вигляді:

$$P_h \approx C_r^{h+1} \cdot (1-g)^{r-h-1} \cdot g^{h+1} \approx \frac{r^{h+1}}{(h+1)!} \cdot g^{h+1} - \frac{r^{h+2}}{(h+1)!} \cdot g^{h+2} .$$

Відповідно, вважаючи, що виникнення помилок в пакетах відбувається незалежно, середня кількість k_h пакетів, які повторно передаються визначається наступною формулою:

$$k_h = n \cdot P_h = \frac{V}{r} \cdot P_h = V \cdot \frac{r^h \cdot g^{h+1}}{(h+1)!} \cdot (1 - r \cdot g) .$$

Тобі загальний час t_T відновлення за рахунок резервних, пакетів, які затримались понад критичний час визначається наступною формулою:

$$t_T = k_h \cdot (\varepsilon - 1) \cdot r \cdot \tau_{XOR} = V \cdot r^{h+1} \cdot g^{h+1} \cdot \frac{1 - r \cdot g}{(h+1)!} \cdot (\varepsilon - 1) \cdot \tau_{XOR} .$$

Аналіз формули (2.м) показує, що найбільш дієвим шляхом зменшення часу відновлення затриманих передачею пакетів є зменшення їх довжини.

Відповідно, що середній час T_m відновлення блоку даних при зберіганні інформації в хмарах визначається формулою:

$$T_m = \varepsilon \cdot t_T + (\varepsilon - 1) \cdot t_{XOR} ,$$

Основним чинником скорочення витрат часу на відновлення даних при їх віддаленому зберіганні, що задіяний в запропонованому методі, є зменшення кількість невтрачених блоків, які потрібно транспортувати по мережі для відновлення втрачених блоків. Аналіз показує, що саме час транспортування по мережі вказаних блоків визначає час реконструювання втраченої інформації. В порівнянні з часом транспортування, саме відновлення, яке реалізується простими логічними операціями займає на порядки менше часу. Якщо виключити втрати інформації цілих сховищ, зумовлені цілеспрямованими хакерськими атаками та військовими і природними катаклізмами, то згідно з [35] основним чинником скорочення витрат часу на відновлення даних при їх віддаленому зберіганні, що задіяний в запропонованому методі, є зменшення кількість невтрачених блоків, які потрібно транспортувати по мережі для відновлення втрачених блоків. Аналіз показує, що саме час транспортування по мережі вказаних блоків визначає час реконструювання втраченої інформації. В

порівнянні з часом транспортування, саме відновлення, яке реалізується простими логічними операціями займає на порядки менше часу.

В роботі розглядається проблема відновлення не спотвореної, а саме втраченої інформації. Тобто мова йде про те, що блок даних користувача, який зберігається на віддаленому сховищі втрачено в результаті технічних несправностей, помилок програмного забезпечення, вірусних атак, чи катаклізмів. Або мова може йти про тимчасову недоступність для користувача його інформації внаслідок перевантаженості системи, проведення ремонтно-профілактичних робіт. З позиції користувача факт втрати блоку фіксується у вигляді відмови доступу до нього. Тобто в рамках дослідження не розглядається ситуація часткової втрати інформації блоку. Проблемі відновлення частково спотворених даних присвячено досить багато досліджень [36,37], починаючи від класичних завадостійких кодів. В системах віддаленого зберігання даних ця задача вирішується на рівні сховища, окремого сектору на диску тощо.

Якщо говорити в термінах, прийняти завадостійкого кодування, то там не враховуються такі важливий критерій ефективності як час транспортування даних до обчислювальної платформи, на якій здійснюється процес відновлення. Тобто, теорія завадостійкого кодування є надто загальною для того, щоб врахувати специфіку конкретної нової області застосування, якою є віддалене зберігання інформації на віддалених сховищах з використанням хмарних технологій.

Щодо підвищення функціональної надійності віддаленого зберігання блоків, що саме рознесення інформації користувача по різним сховищам здатне забезпечити найвищий рівень надійності. Це зумовлене тим, що рознесення інформації по різним сховищам виключає можливість одночасної втрати значної частити даних. Максимально, що може користувач одночасно втратити – це блоки, які зберігаються на одному сховищі. Зрозуміло, що ефективність рознесення зберігання даних забезпечується тільки за наявності засобів ефективного і оперативного відновлення втраченої в рамках одного сховища інформації. Типовою системою подібного типу є RAID. Якщо втрачається вся

інформація на сховищі внаслідок, наприклад цілеспрямованої хакерської атаки, вірусних програм чи катаклізмів природного чи військового характеру, то існуючі системи, включаючи RAID не здатні відновити втрачені дані.

При організації відновлення на рівні користувача, інформаційні та резервні блоки котрого зберігаються на великій кількості сховищ, втрата їх частини на одному сховищі не є фатальною: втрачені дані можуть бути відновлені за рахунок інформації користувача на інших сховищах.

Практично всі методи складаються з двох процедур. Перша з них – це процедура формування резервних блоків, а друга – це відновлення втрачених блоків з використанням невтрачених інформаційних та резервних блоків. Перша процедура полягає в способі побудови матриці формування резервних блоків виходячи із поставленої задачі. В розробленому методі матриця формування резервних блоків будується з використанням пріоритетів окремих інформаційних блоків. Друга процедура – це власне процес відновлення втраченої інформації. Цей процес включає віднаходження в підматриці, стовпці якої співвідносяться з втраченими інформаційними блоками, а рядки – з невтраченими резервними блоками, ортогональної підматриці. Це дозволяє скласти систему лінійних рівнянь, розв’язання якої задає порядок відновлення втрачених блоків.

Інформаційна надлишковість запропонованого методу близька до теоретичного мінімуму, тобто до рівня, який досягається в класичних MDS кодах. Наприклад, якщо кількість n інформаційних блоків дорівнює 12-ти, і гарантовано відновлюються чотири блоки, то число резервних блоків становить 10. Приклад такої матриці формування резервних блоків:

$$A_c = \begin{vmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{vmatrix}$$

Для того, щоб забезпечити ортогональність підматриць можна використовувати системи квазістаціонарних рекурентних функцій Галуа. Ці функції базуються на ортогональних систем функцій базису Уолша. Системи дискретних булевих функцій Уолша являють собою системи ортонормованих прямокутних бінарних функцій. Розглядаються різні способи впорядкування систем такого класу. При цьому з базису Уолша трансформуються в базиси Галуа чи Адамара. На основі останніх можуть бути побудовані перевірочні матриці широкого класу корегуючих функцій, таких зокрема, як коди Хеммінга та циклічні коди.

Для задачі гарантованого відновлення інформаційних блоків при втраті не більше h із них потрібні непрямокутні матриці

Прикладом системи А, яка гарантує відновлення всіх інформаційних пакетів при втраті не більше 4-х з $n+k$ переданих при n=14 та k=10 є система, перший рядок якої утворений послідовністю S(n), а інші рядки – циклічним зсувом вказаної послідовності.

Послідовність S(n) утворюється бітами переносу зсувного 5-роздядного регістру зі лінійною функцією зворотного зв’язку, яка забезпечує період повторення n=14. Якщо позначити через R(j) поточний код на регістрі на j–тому кроці, а через F – код функції зворотного зв’язку, то послідовність формування S(n) може бути описана наступним чином:

$$\forall j \in \{1,2,\ldots,n\}:\ R(j) = R(j\text{-}1)\cdot 2 + s_j\ ,\ s_j = r_1(j\text{-}1)\cdot f_1 \oplus r_2(j\text{-}1)\cdot f_2 \oplus \ldots \oplus r_4(j\text{-}1)\cdot f_4$$

Початкове значення $R(0) = 30_{10} = 11110$ $F = x^6 + x^5 + x^4 = 11100$

Формування послідовності ілюструється наступною таблицею

Крок j	R	$s_j = r_5 \oplus r_4 \oplus r_3$	Крок j	R	$s_j = r_5 \oplus r_4 \oplus r_3$
0	1 1 1 1 0	1	7	0 0 0 0 1	0
1	1 1 1 0 1	1	8	0 0 0 1 0	0
2	1 1 0 1 1	0	9	0 0 1 0 0	1
3	1 0 1 1 0	0	10	0 1 0 0 1	1
4	0 1 1 0 0	0	11	1 0 0 1 1	1
5	1 1 0 0 0	0	12	0 0 1 1 1	1
6	1 0 0 0 0	1	13	0 1 1 1 1	0

Число компонентів R	Довжини циклів, для яких існує лінійна функція зворотного зв’язку
4	5, 6, 7, 15
5	5, 6, 7, 8, 12, 14, 15, 21, 31
6	5, 6, 8, 9, 10, 12, 14, 15, 21, 28, 30, 31, 47

Побудована матриця А

$$A = \begin{vmatrix} 1011000111 \\ 0110001111 \\ 1100011110 \\ 1000111101 \\ 0001111011 \\ 0011110110 \\ 0111101100 \\ 1111011000 \\ 1110110001 \end{vmatrix}$$

При двох інформаційних блоках, n=2 для гарантованого їх відновлення за умови, що буде втрачено не більше 4-х блоків, кількість k резервних блоків має становити шість. Дійсно, оскільки в кожному з двох стовпців матриці А має міститися не менше 4-х одиниць, то кількість рядків не може бути меншою чотирьох. В силу того, що Хемінгова відстань між цими стовпцями має бути не

менше трьох, то кількість нулів в стовпцях матриці А не може бути меншою 2-х. Відповідно, мінімальна кількість *k* рядків матриці А має дорівнювати 6-ти, тобто *k*-6. Матриця А може мати наступний вигляд:

$$A = \begin{vmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{vmatrix}$$

При *n*=3 для вирішення задачі гарантованого відновлення інформаційних блоків, коли загальна кількість втрачених не перевищує 4-х підбір матриці А, що містить лише шість рядків і задовольняє сформульованим вище вимогам не можливим. Покажемо це наступним чином. Без втрати узагальнення можна вважати, що перший стовпчик ζ_1 матриці А складається з 4-х одиниць і двох нулів, тобто має вигляд: ζ_1 = { 1,1,1,1,0,0}, другий стовпчик ζ_2 матриці А також має містити 4 одиниці і різнитися від першого не менш ніж в трьох розрядах. Єдиним варіантом виконати вказане є здійснення вибору ζ_2 так, щоб його два нулі знаходилися в тих рядках, в яких стовпчик ζ_1 має одиниці, тобто ζ_2 = { 1,1,0,0,1,1}. Третій стовпчик ζ_3 матриці А має бути вибраним таким чином, щоб він містив не менше 4-х одиниць, знаходився на Хемінговій відстані від стовпців ζ_1 і ζ_2 не менше 3-х і утворював зі згаданими стовпцями ортогональну підматрицю Θ. Для виконання останньої умови, матриця А має містити не менше двох рядків з непарною кількістю одиниць. Якщо перший рядок матриці А містить всі одиниці, а другий – дві одиниці і нуль, то в Хемінгова відстань між чотирма останніми компонентами ζ_3, серед яких лише один нуль, та відповідними фрагментами ζ_1 і ζ_2 має бути не менше 2-х, що, вочевидь, неможливо. Якщо перші дві компоненти стовпчика ζ_3 дорівнюють нулю, а інші чотири одиниці, то матриця А не містить рядків з непарною кількістю одиниць, а це значить, що в ній нема ортогональної підматриці Θ.

Таким чином, при n=3 для гарантованого відновлення всіх інформаційних блоків за умови втрати не більше чотирьох з усіх, що передаються, потрібно не менше 7-ми резервних блоків. При цьому матриця А, що описує формування резервних блоків може мати такий вигляд:

$$A = \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}. \qquad (2.7)$$

В матриці А (2.7) виконуються всі умови для відновлення інформаційних блоків при втраті не більше 4-х блоків з числа інформаційних і резервних. Зокрема, матриця А містить три рядка з непарною кількість одиниць, що дозволяє гарантовано відновлювати три інформаційних блоки при їх втраті разом з одним із резервних. Наприклад, якщо крім трьох інформаційних втрачено ще перший резервний, то інформаційні можуть бути відновлені з використанням ортогональної підматриці Θ, що міститься в матриці А:

$$\theta = \begin{vmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

Відповідно, процес відновлення здійснюється у наступному вигляді:

$$B_2 = Q_6$$

$$B_3 = Q_7$$

$$B_1 = Q_3 \oplus B_3$$

Якщо число інформаційних блоків дорівнює чотири, тобто n=4, всі вони за умови втрати не більше 4-х блоків можуть бути гарантовано відновлені з використанням семи резервних блоків (k=7). Матриця А, що описує формування резервних блоків може бути отримана з матриці (2.4) додаванням четвертого стовпця так, щоб виконувалися зазначені умови:

$$A = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{vmatrix}$$

Матриця А задовольняє всім умовам для гарантованого відновлення всіх інформаційних блоків при втраті не більше чотирьох блоків з множини інформаційних та резервних. Оскільки будь-яка підмножина трьох з чотирьох стовпців матриці А містить два рядки з непарною кількістю одиниць, то це гарантує відновлення будь-яких трьох інформаційних блоків при втраті крім них будь-якого одного резервного.

При втраті 4-х інформаційних, їх відновлення гарантується наявністю в матриці А ортогональної підматриці Θ:

$$\theta = \begin{vmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{vmatrix}$$

Процес відновлення втрачених інформаційних блоків B_1,B_2,B_3,B_4 здійснюється шляхом використання наступних обчислень:

$$B_1 = Q_3 \oplus Q_6 \oplus Q_7$$

$$B_2 = Q_2 \oplus Q_3 \oplus Q_6 \oplus Q_7$$

$$B_3 = Q_2 \oplus Q_3 \oplus Q_6$$

$$B_4 = Q_2 \oplus Q_3 \oplus Q_7$$

При п’яти інформаційних блоках (n=5) для того, щоб гарантувати відновлення будь-яких чотирьох із них за умови, що втрачено не більше чотирьох з загальної кількості інформаційних та резервних блоків, мінімальне число k резервних блоків становить сім: k=7. Матриця А формування резервних блоків при цьому має наступний вигляд:

$$A = \begin{vmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{vmatrix}. \quad (2.8)$$

Легко переконатися, що матриця (2.8) задовольняє всім вимогам: кожен з 5-ти її стовпців містить щонайменше 4 одиниці, Хемінгова відстань між будь якою парою стовпців не менша 3-х, підматриця матриці утворена будь-якою трійкою стовпців матриці А містить ортогональну квадратну підматрицю, підматриця, утворена будь-якою четвіркою стовпців також містить ортогональну підматрицю.

Зокрема, якщо втрачені чотири перші інформаційні блоки, то підматриця, складена з 4-х перших стовпців матриці А (2.8) містить в собі ортогональну підматрицю Θ:

$$\Theta = \begin{vmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{vmatrix}$$

Відповідно, відновити інформаційні блоки B_1-B_4 можна за формулами, які легко отримуються як рішення системи рівнянь, яка. в свою чергу, будується на основі ортогональної матриці Θ:

$$\begin{cases} Q_1 = B_1 \oplus B_4 \\ Q_2 = B_2 \oplus B_3 \oplus B_4 \\ Q_5 = B_1 \oplus B_2 \oplus B_4 \\ Q_7 = B_2 \oplus B_4 \end{cases}. \quad (2.9)$$

Рішення системи (2.9) формують формули для відновлення втрачених інформаційних блоків:

$$B_1 = Q_5 \oplus Q_7$$

$$B_2 = Q_1 \oplus Q_5$$

$$B_3 = Q_3 \oplus Q_7$$

$$B_4 = Q_1 \oplus Q_5 \oplus Q_7$$

При шести інформаційних блоках мінімальна кількість резервних блоків для гарантування відновлення втрачених інформаційних, за умови не більше чотирьох втрачених (інформаційних і резервних) блоків також дорівнює шести.

Для цього резервні блоки можуть формуватися у відповідності з наступною матрицею А:

$$A = \begin{vmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{vmatrix}$$

Наприклад, при втраті двох резервних блоків, наприклад першого і другого, та двох інформаційного, також першого B_1 і другого B_2, вони можуть бути відновлені шляхом наступних обчислень:

$$B_1 = Q_4 \oplus B_4 \oplus B_6$$

$$B_2 = Q_7 \oplus B_4 \oplus B_6$$

$$A = \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

Приклад системи, яка гарантує відновлення будь-яких чотирьох втрачених блоків з загальної кількості 6-ти інформаційних блоків. Система використовує 7 резервних блоків:

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Для конструювання матриці були вибрані стовпці, що мають чотири одиниці і три нуля, а також стовпець, що складається з одиниць.

2.2. Розробка процедур формування резервних блоків

Пришвидшення процедур обрахунку для утворення блоків резервного копіювання завдяки встановлення ліміту (до 3 блоків) на кількість блоків, які можливо відновити є метою дослідження.

Щоб її досягнути потрібно розв’язати наступні поставлені наукові завдання:

- дослідити теоретично особливості, притаманні надлишковим блокам, що надають гарантію для відновлення блоків інформації у випадку, якщо втратили кількість, що не перевищує 3 від сумарної кількості;
- розробити ґрунтуючись на одержаних результатах після теоретичних досліджень, методу для формування блоків резервного копіювання, щоб відновити блоки інформації гарантовано за втрати, що не перевищує 3-ох блоків від сумарної кількості;

Далі буде описана модель, яка лежить в основі запропонованого методу: дані, що пересилаюся через мережу мають вигляд групи з n основних блоків $P_1, P_2, \dots, P_n$. Кожен блок вміщає коди контролю, за допомогою яких можна локалізувати помилку.

За замовчуванням трафік впливає на те, якими шляхами буде відправляти блок. До того ж, шлях може містити ділянки однорангових мереж (P2P – peer-to -peer). Що стосується цих ділянок, то є можливість неконтрольованого

неочікуваного вимкнення, під час якого порушується дієздатність каналу передачі на деякий проміжок часу, коли можлива втрата блоків інформації.

Також зазвичай береться до уваги характеристика моменту отримання блоків, тобто блок інформації актуальний впродовж визначеного критичного часу.

Тому при втраті розглядають три варіанти можливих випадків:

- помилки неможливо виправити з використанням кодів контролю, що включені до вмісту блоку;
- блок втратили при передачі;
- блок надійшов до назначеного вузла після того, коли інформація, що міститься в ньому, втратила актуальність.

Щоб організувати процес відновлення блоків, що були втрачені при передачі, запропоновано додатково до основних блоків відправляти блоки резервного копіювання, що утворюються певним способом з використанням вмісту основних блоків.

Для збільшення швидкості відновлення основних блоків, що були втрачені, представляється спосіб утворення блоків резервного копіювання, при якому відновлення відбувається за допомогою лінійних булевих перетворень, перевагою використання яких є проста апаратна імплементація, яка полягає в тому, що блок резервного копіювання утворюється обрахунком суми за модулем два кодів конкретної множини з основних блоків.

У вузлі призначення у випадку, коли деяка кількість основних блоків не були отримані до визначеної часової межі, їх приймають як втрачені. Їх відновлення можливе за допомогою лінійних перетворень, що виконуються із використання основних блоків та блоків резервного копіювання, які прийшли до критичного часу.

Якщо втрачено три блоки із загальної кількості, то потрібно розглянути чотири випадки:

1. втрачені блоки належать до основних блоків;

2. два втрачені блоки належить до основних, один – до блоків резервного копіювання;
3. два втрачені блоки належать до блоків резервного копіювання, один – до основних;
4. втрачені блоки належать до блоків резервного копіювання.

У четвертій ситуації всі втрати приходяться на блоки резервного копіювання, а отже всі основні блоки досягли вузлу призначення, тому відновлення не потребують.

У третій ситуації потрібно відновити один основний блок. Відновлення можливе тільки, якщо матриця A не вміщатиме стовпчиків з нулями після того, як викреслимо рядки матриці, що відповідають втраченим блокам.

Щоб виконувалася описана вище умова достатньо того, щоб кожен стовпець матриці мав більше, ніж дві одиниці:

$$\forall l \in \{1,2,...,n\} : \sum_{i=1}^{m} a_{i,l} > 2 \tag{2.4}$$

У другій ситуації, щоб відновити втрачені основні блоки, потрібно отримати різні стовпці після закреслення з матриці рядків, що відповідають втраченим блокам. Для виконання зазначеної умови необхідно, щоб відстань Хемінга між стовбцями у матриці становила більше 1:

$$\forall l, i \in \{1,2,...,n\}, l \neq i : \sum_{j=1}^{m} (a_{j,l} \oplus a_{j,i}) > 1 \tag{2.5}$$

У першій ситуації за виконання умов (2.4) та (2.5) блоки, що втратили при передачі, можливо відновити гарантовано, якщо виконується умова, що один рядок містить тільки одиниці:

$$\exists j \in \{1,2,...,m\} : \prod_{i=1}^{n} a_{j,i} = 1 \tag{2.6}$$

Доведення. Позначимо номери блоків a, b, c, де $a, b, c \in \{1, 2, \dots, n\}$ та $a < b < c$. Тому матрицю Δ можна представити зі стовбців матриці A як:

$$\Delta = \begin{vmatrix} a_{1,v} & a_{1,w} & a_{1,e} \\ a_{2,v} & a_{2,w} & a_{2,e} \\ & \cdots & \\ a_{m,v} & a_{m,w} & a_{m,e} \end{vmatrix}$$

Відновлення буде успішним за умови наявності у матриці Δ трьох рядків, з яких ортогональна матриця θ може бути сформована.

Щоб довести вище викладене необхідно провести дослідження диференційних характеристик, що притаманні рядкам матриці Δ. Дані характеристики відповідають дворозрядному векторові $d_y = \langle d_1, d_2 \rangle, y \in \{1,2, \dots, m\}$, y – номер рядку у матриці Δ:

$$d_y = \langle d_1, d_2 \rangle, \text{де } d_1 = a_{ya} \oplus a_{yb}, d_2 = a_{yb} \oplus a_{yc}.$$

За формулою (2.4) відстань Хемінга між 1-им і 2-им стовбцями матриці Δ не перевищує двох, що значить матриця Δ містить як мінімум два рядки, перша складова вектору d для яких рівна 1: $d = \langle 1,0 \rangle$ чи $d = \langle 1,1 \rangle$. Одночасно з цим для кожного значення вектору d відповідно є два варіанти рядку в матриці Δ і можливо два випадки:

1. у двох рядках 1-ий та 2-ий стовпці являються інверсією, тобто беруться відносно одного значення d. Наприклад, інверсними рядками є 010 і 101 відносно $d = \langle 1,0 \rangle$;
2. рядки беруться відносно різних значень d, а отже вони не інверсні. Наприклад, рядок 101 береться відносно $d = \langle 1,0 \rangle$, а рядок 011 – відносно $d = \langle 1,1 \rangle$.

Для кожного значення вектору d знайдеться два рядки у матриці Δ, а тому у матриці Δ буде як мінімум 2 рядки, що входять у множину $Y_1 = \{\langle 0,1,0 \rangle; \langle 1,0,1 \rangle; \langle 0,1,1 \rangle, \langle 1,0,0 \rangle\}$.

Аналогічним чином у матриці Δ міститься як мінімум 2 рядки, які різняться 1-им та 3-ім стовпцями, що означає рівність другої складової вектору d одиниці: $d = \langle 0,1 \rangle$ чи $d = \langle 1,1 \rangle$. Рядки, що різняться значенням 1-ої і 3-ої складових, можуть братися відносно однаковим або різним значенням d. Це, в свою чергу,

означає, що матриця Δ містить як мінімум 2 рядки, що можуть входити до множини $Y_2 = \{\langle 0,0,1\rangle; \langle 1,1,0\rangle; \langle 0,1,1\rangle, \langle 1,0,0\rangle\}$.

Керуючись тією ж аналогією, можна зазначити, що матриця Δ містить як мінімум 2 рядки, 2-ий та 3-ій стовпчики яких різняться: $d = \langle 0,1\rangle$ чи $d = \langle 1,0\rangle$. Вони можуть братися відносно до одного значення d, тобто бути інверсно відображені, або ж братися відносно двох значень d, тобто бути не інверсними. Тоді матриця Δ містить як мінімум 2 рядки, що будуть входити до множини $Y_3 = \{\langle 0,0,1\rangle; \langle 1,0,1\rangle; \langle 0,1,0\rangle, \langle 1,1,0\rangle\}$.

Якщо виконати перетини пар вищезазначених множин Y_1, Y_2, Y_3, то можна побачити, що кожен має по дві складові. Якщо ж виконати перетин всіх множин результатом має порожню множину: $Y_1 \cap Y_2 \cap Y_3 = \emptyset$. Для виконання умови необхідно, щоб матриця Δ вміщала по два рядки з кожної множин Y_1, Y_2, Y_3.

За умови, що матриця Δ вміщає два рядки з множини Y_1 і вони не інверсні між собою, можна сказати, що ці два рядки взяті відносно різних значень d. Інакше, коли вони інверсні – їх взято по відношенню до однакового значення d_1 і містяться в множині Y_2 чи Y_3. За умови, що згадані два рядки містяться у Y_1 та Y_2, повинно бути додатково пара рядків, що містяться у Y_3 і такі, що $d_1 \neq d_2$, щоб виконувалася умова (2.4). Аналогічно для випадку, при якому пара рядків міститься у множинах Y_1 та Y_3.

Підсумовуючи вищезгадане: щоб виконувалася умова (2.5), матриця Δ повинна містити рядки, що були б взяті відносно 2-ох різних значень d.

Можливі комбінації пар рядків, що утворюють матрицю Δ та взяті відносно різних значень d внесене до таблиці 2.1. За умовою (2.5) у таблицю внесено одиничний ряд, що міститься у матриці Δ, до пар.

Таблиця 2.1

Варіанти ортогональних під матриць з матриці Δ для різних комбінацій пар значень d

Комбінації пар векторів d	Комбінації рядків відповідно до значень d			
$\langle 0,1\rangle, \langle 1,0\rangle$	$\begin{vmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{vmatrix}$	$\begin{vmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{vmatrix}$	$\begin{vmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{vmatrix}$	$\begin{vmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{vmatrix}$
$\langle 0,1\rangle, \langle 1,1\rangle$	$\begin{vmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}$	$\begin{vmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{vmatrix}$	$\begin{vmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}$	$\begin{vmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{vmatrix}$
$\langle 1,0\rangle, \langle 1,1\rangle$	$\begin{vmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}$	$\begin{vmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{vmatrix}$	$\begin{vmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}$	$\begin{vmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{vmatrix}$

Всі занесені до таблиці 2.1 матриці являються ортогональними. Вони сформовані з рядків, що взяті відносно різних значень d та одиничним рядком.

Отже, проаналізувавши таблицю 2.1 можна прийти до наступних висновків: за умови, що матриця Δ вміщає одиничний рядок та рядки, взяті відносно різних значень вектору d, в матриці Δ буде вміщати мінімум одну ортогональну під матрицю, з чого доведено можливість відновлення трьох блоків, що були втрачені.

У запропонованому методі утворення m блоків резервного копіювання $R_1, R_2, \dots, R_m$ відбувається за допомогою лінійних комбінацій відповідних розрядів. Звідси утворення блоків резервного копіювання переходить у задачу ефективно побудувати матрицю A, що відповідала б умовам (2.4), (2.5), (2.6).

Для виконання умови (2.4) відстань Хемінга між довільними двома стовпцями матриці A повинна бути рівною мінімально двом. У той же час для виконання умови (2.5) число «1» у всіх стовпцях матриці повинне бути рівним мінімально трьом. Тоді кандидатами на місце стовпця для формування матриці може бути n-бітові двійкові коди, у яких є 3, 5,7, …, m «1».

Відстань Хемінга рівна 2 для двійкових кодів, що є різними та містять однакову кількість «1». Відстань Хемінга не менша 2-ох у випадку, коли кількість одиниць у двійкових кодах відрізняється мінімум на два.

Кількість n усіх можливих варіацій стовпчиків, у яких є 3, 5,7, …, m «1» відповідно до (2.6) та обставини наявності одного одиничного рядка в матриці A можна визначити як суму числа перестановок відповідно трьох, п’яти, семи і аналогічно «1» в $(m-1)$-бітовому двійковому коді:

$$n = \sum_{j=1}^{v} C_{m-1}^{2-j}, \text{де} \begin{cases} v = \frac{m-1}{2} \text{ при } m\%2 = 1 \\ v = \frac{m}{2} - 1 \text{ при } k\%2 = 0 \end{cases} \quad (2.7)$$

Застосувавши властивості біноміальних коефіцієнтів:

$$\sum_{i=0}^{m} C_m^i = 2^m, \quad \sum_{i=0}^{m} (-1)^i \cdot C_m^i = 0,$$

можливо привести формулу (2.7) до виду:

$$n = \sum_{j=1}^{\upsilon} C_{m-1}^{2 \cdot j} = 2^{m-1} - 1. \quad (2.8)$$

За допомогою формули (2.8) можливо обчислити кількість m блоків резервного копіювання, які потрібні, щоб відновити три блоки з n основних блоків:

$$m \geq \lfloor \log_2 (n+1) \rfloor + 2. \quad (2.9)$$

Розглянемо приклад, у якому число n = 12 основних блоків, тоді, щоб відновити 3 втрачені блоки необхідно додатково відправити m = 6 блоків резервного копіювання. Отже, матриця A утворення блоків резервного копіювання буде виду:

$$A = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{vmatrix}$$

Якщо збільшувати число n основних блоків, то збільшується і ефективність створення резервних копій. Так, наприклад, за числа основних блоків n =120 число блоків резервного копіювання буде m = 9. Як видно, це становить 7,5% від числа основних блоків.

Можна навести такий приклад процесу відновлення втрачених основних блоків з інформацією при передачі глобальною мережею. У прикладі буде використано вищезазначену матрицю A за умови передачі n = 12 основних блоків та m = 6 блоків резервного копіювання.

Нехай при передачі було втрачено два основні блоки v =5 та w = 7 і один блок резервного копіювання q = 4.

Тоді отримуємо таку підматрицю Δ матриці A, яка формується за допомогою стовпчиків відповідно до втрачених основних блоків та рядків відповідно з невтраченими блоками резервного копіювання:

$$\Delta = \begin{vmatrix} a_{1,5} & a_{1,7} \\ a_{2,5} & a_{2,7} \\ a_{3,5} & a_{3,7} \\ a_{5,5} & a_{5,7} \\ a_{6,5} & a_{6,7} \end{vmatrix} = \begin{vmatrix} 1 & 1 \\ 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 1 \end{vmatrix}$$

Можна побачити, що матриця Δ має підматрицю θ, що є ортогональною та утворюється за участі 3-ого та 5-ого рядків з матриці Δ:

$$\Theta = \begin{vmatrix} a_{3,5} & a_{3,7} \\ a_{6,5} & a_{6,7} \end{vmatrix} = \begin{vmatrix} 1 & 1 \\ 0 & 1 \end{vmatrix}$$

Далі замість третьої та шостої компонент блоків резервного копіювання застосуємо Y_3 та Y_6, Q_3 та Q_6 відповідно. Тоді отримуємо систему з лінійних рівнянь:

$$\begin{cases} Y_3 = P_5 \oplus P_7 \\ Y_6 = P_7 \end{cases}$$

Розв’язавши її ми отримаємо формули, за допомогою яких можемо відновити втрачені блоки:

$$\begin{aligned} P_7 &= Y_6 \\ P_5 &= Y_3 \oplus Y_6 \end{aligned}. \qquad (2.10)$$

Для відновлення двійкових кодів основних блоків, що були втрачені у процесі передачі, необхідно обчислити складові Y_3 та Y_6 блоків резервного копіювання. Вони утворюються за допомогою суми по модулю 2 відповідних складових матриці A:

$$Q_3 = P_1 \oplus Y_3 \oplus P_6 \oplus P_9 \oplus P_{10}$$
$$Q_6 = P_4 \oplus Y_6 \oplus P_9 \oplus P_{10} \oplus P_{12}$$

Звідси, компоненти Y_3 та Y_6 блоків резервного копіювання можна відобразити так:

$$Y_3 = Q_3 \oplus P_1 \oplus P_6 \oplus P_9 \oplus P_{10}$$
$$Y_6 = Q_6 \oplus P_4 \oplus P_9 \oplus P_{10} \oplus P_{12}$$

Якщо підставити ці значення для Y_3 та Y_6 у формулу (2.10), щоб відновити 5-ий та 7-ий основні втрачені блоки, одержимо вирази, які можна використати для відновлення цих блоків. Відновлення відбувається з використанням основних блоків та блоків резервного копіювання, що надійшли до вузлу призначення вчасно та коректно:

$$P_7 = Y_6 = Q_6 \oplus P_1 \oplus P_6 \oplus P_9 \oplus P_{10}$$
$$P_5 = Y_6 \oplus Y_3 = Q_3 \oplus Q_6 \oplus P_1 \oplus P_4 \oplus P_6 \oplus P_{12}$$

Підводячи підсумок, для відновлення 5-ого та 7-ого основних втрачених блоки можна використати лінійні перетворення над основними блоками та блоками резервного копіювання, що надійшли без затримок та спотворень.

2.3. Оцінка ефективності

З теоретичної точки зору вірогідність того, що можна відновити блоки інформації, число яких менше або рівне *k*, сформульовано може бути так: зрозуміло, нам не вдасться відновити блоки за умови, що число лінійно незалежних рівнянь, в які перетворюються віднімання між складовими коду керування відправника та одержувача, менше k. Цей випадок виникає, якщо місце всіх розрядів блоку, які спотворено отримані, за своїм номером розрізняються тільки на $k - 2$ чи менше біт. Тобто система $k - 1$ незалежних лінійних рівнянь формується з віднімань між складовими коду керування відправника та одержувача. Цю систему не можна використати для локалізації розрядів блоку інформації, що передалися недостовірно.

Вірогідність того, що такий випадок станеться у теорії можна розрахувати взявши у роздуми наступне: нехай q чисел розрізняються на місці d бітів, тоді будемо мати нерівність:

$$\lfloor \log_2 q \rfloor \leq d \leq q - 2.$$

З цього зрозуміло, що число всіх випадків Q_1 становитиме $s = 2^d : C_s^q$, а число Q_2, яке відповідає за випадки розташування q-бітових кодів у n-бітовому номері, рівне C_n^q. Виходячи з цього, сумарне число випадків різних чисел за кодом по розрядах за умови різниці тільки у q розрядах, вираховується шляхом множення $Q_1 * Q_2$. Сумарна кількість Q_0 випадків, коли обираємо d n-бітові номери, дорівнює $b = 2^n : C_b^d$. Отже, можна визначити вірогідність P_0, яка відноситься до випадків, коли d n-бітові номерів відрізняються менше, ніж $d - 1$ бітів:

$$P_0 = \sum_{q=\lfloor \log_2 d \rfloor}^{d-2} \frac{n!}{(n-q)! * q!} * \prod_{i=0}^{d-1} \frac{(2^q - i)}{(2^n - 1)}.$$

Від зворотного, вірогідність P_1 відновлення d інформаційних блоків: $P_1 = 1 - P_0$.

У ході роботи було розроблено програму та на її основі зроблено емпіричні досліди, щоб перевірити викладені вище теоретичні положення. Таблиця 2.1 вміщає результати емпіричних дослідів.

Таблиця 2.2

Вірогідність, що відновимо втрачені блоки

Число основних блоків *n*	Число блоків резервного копіювання *k*	Число блоків, що втратили			
		2	3	4	5
5	2	0,81002	-	-	-
	3	0,85654	0,80301	-	-
7	2	0,83333	-	-	-
	3	0,86667	0,83941	-	-
	4	0,72697	0,58692	0,84014	-
10	2	0,84908	-	-	-
	3	0,77013	0,82495	-	-
	4	0,83499	0,68093	0,84916	-
	5	1,0	0,84837	0,79519	0,77941
12	2	0,84595	-	-	-
	3	0,81896	0,84902	-	-
	4	0,81701	0,73988	0,878413	-
	5	0,99301	0,86948	0,845991	0,83969
16	2	0,86300	-	-	-
	3	0,80120	0,85992	-	-
	4	0,83161	0,80938	0,90851	-
	5	0,91939	0,89006	0,89156	0,89993
24	2	0,85982	-	-	-
	3	0,79043	0,86655	-	-
	4	0,77239	0,79992	0,90198	-
	5	0,77601	0,79867	0,86812	0,92495
32	2	0,84981	-	-	-
	3	0,79913	0,88046	-	-
	4	0,78609	0,83961	0,91923	-
	5	0,87145	0,85884	0,92957	0,93758
64	2	0,88570	-	-	-
	3	0,81682	0,89619	-	-
	4	0,76929	0,85982	0,92057	-
	5	0,80926	0,86954	0,93506	0,94592

Вірогідність відновити блоки з інформацією також порівнювалася з застосуванням методу, що розробили, для формування блоків резервного копіювання і утворення блоків резервного копіювання рандомно, за тим, як виконувалося до цього. В таблицю 2.2 вміщено результати.

Таблиця 2.3

Вірогідність відновити блоки, що втрачено, з використанням додаткових блоків, що утворені за допомогою запропонованого методу та рандомно

Число основних блоків *n*	Число блоків резервного копіювання *k*	Число блоків, що втратили		
		3	4	5
16	2	-	-	-
	3	0,77982 (-0,07956)	-	-
	4	0,71083 (-0,10105)	0,82743 (-0,09394)	-
	5	0,80299 (-0,08597)	0,80943 (-0,07849)	0,84924 (-0,04602)
24	2	-	-	-
	3	0,75198 (-0,11593)	-	-
	4	0,75175 (-0,05816)	0,70398 (-0,19526)	-
	5	0,76951 (-0,02899)	0,84497 (-0,02765)	0,87824 (-0,04495)
32	2	-	-	-
	3	0,79066 (-0,09098)	-	-
	4	0,70252 (-0,14286)	0,87492 (-0,05673)	-
	5	0,79394 (-0,07009)	0,81156 (-0,11263)	0,87595 (-0,06041)
64	2	-	-	-
	3	0,81295 (-0,09377)	-	-
	4	0,72426 (-0,137082)	0,84062 (-0,09094)	-
	5	0,80825 (-0,06749)	0,85820 (-0,08194)	0,88931 (-0,054881)

Якщо порівняти вірогідності при тому, що ми відновлюємо втрачені блоки використовуючи блоки резервного копіювання, то усереднено на 8,37% ефективність вище за умови утворення додаткових блоків методом, що вище запропонували, ніж якби вони було утворені рандомно.

Після виконання дослідів стало можливим сформулювати кроки для того, щоб формувати додаткові блоки, які б дозволили відновлювати втрачені основні блоки найбільш вірогідно.

Згадані кроки утворення блоків резервного копіювання містять наступні кроки:

1. Застосовуючи таблицю 2.1 визначити потрібне число *k* резервних блоків орієнтуючись такими параметрами:
   - числом n основних блоків,
   - необхідною вірогідністю, щоб відновити втрачені блоки,
   - числом блоків, що були втрачені, прийшли спотворені або невчасно.
2. За умови $2^k > n$: вибираємо n стовпці $c_1, c_2, \dots, c_n$ не повторюючись та за зниженням порядку пріоритетності $\varphi_1 \geq \varphi_2 \geq \cdots \geq \varphi_n$.
3. За умови $2^k - 1 = n$: вибираємо всі наявні $2^k - 1$ стовпці не повторюючись.
4. За умови $2^k \leq n$: тоді n беремо як $n = a * (2^k - 1) + b$, a та b – цілі числа і $b < 2^k - 1$. Група стовпців утворюється повторенням a разів $2^k - 1$ вірогідний k-бітових кодів стовпців $c_1, c_2, \dots, c_n$, плюс беремо b стовпців від стовпця з найбільшим пріоритетом до стовпця з найменшим.
5. Стовпці $c_1, c_2, \dots, c_n$, що утворилися будуть матрицю Ω, складові векторів λ виступають рядками: складова λ_1 – 1-а складова у стовпці, відповідно складова λ_2 – 2-а складова і подібним чином далі для стовпців $c_1, c_2, \dots, c_n$по всій матриці.

Приклад. Нехай n = 10, при передачі втратили 3 блоки, вірогідність відновити рівна 80%.

За таблицею 2.1 блоків резервного копіювання k = 3. В результаті отримуємо таку матрицю Ω, бо $2^3 < 10$:

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
λ_1	1	1	1	0	1	0	0	1	1	1
λ_2	1	1	0	1	0	1	0	1	1	0
λ_3	1	0	1	1	0	0	1	1	0	1

Висновки до розділу 2

У другому розділі роботи представлено досліди, присвячені вирішенню проблеми, щоб відновити ефективно блоки з інформацією, що при передачі втратили, отримали за спотворенням або одержані пізно. Запропонований метод може бути застосований і для даних, що зберігаються на ремоуті щ використанням хмарних засобів збереження. У результаті можна дійти таких висновків:

1. У розділі обґрунтовано та доведено з теоретичної точки зору метод, що запропонований, щоб виправляти багатократні помилки та відновлювати інформацію, що при передачі втратили, отримали зі спотворенням або одержану пізно на рівні блоків. Щоб відновити необхідно використати блоків, що надійшли вчасно та блоків резервного копіювання, утвореним певним чином застосовуючи лінійні функції та які за основу мають основні інформаційні блоки. Викладено описання методу, щоб відновити інформацію.
2. Запропоновано метод, щоб обрати лінійні функції та утворити систему кодів резервного копіювання, використовуючи які можна з максимальною вірогідністю відновити визначене число основних блоків, які втратили, відштовхуючись від числа та способу утворення блоків резервного копіювання.
3. Розроблено програму, що моделює та тестує запропонований метод емпірично. Отримані результати показали, що застосування запропонованого методу є ефективним.
4. Розроблений метод пропонує утворення блоків резервного копіювання як erasure codes, які отримані у результаті суми за модулем 2 кодів з основних блоків. Теоретично та емпірично у роботі довели, що цей підхід надає можливість відновити інформацію ефективніше.

РОЗДІЛ 3

РОЗРОБКА ТЕХНОЛОГІЇ ВІДНОВЛЕННЯ ВТАЧЕНИХ БЛОКІВ ДАНИХ

Технологічні аспекти вирішення систем лінійних рівнянь для визначення процедури відновлення значною мірою впливають на ефективність процесу відновлення та резервування втрачених блоків даних при їх передачі через мережу. Розробка технологій ефективного відновлення даних, втрачених при передачі, є можливою на основі запропонованого методу. Така технологія повинна забезпечити значну швидкодію процесу відновлення інформації за рахунок вибору найбільш оптимального з точки зору обчислювальних операцій варіанту реконструювання, а також використання так званих попередніх обчислень, що уможливлює виключення деяких кроків з процесу обчислення, пов’язаних із вирішенням систем лінійних рівнянь.

Зважаючи на вищесказане, важливість задачі розробки технології реконструювання втрачених блоків інформації за запропонованим методом є вагомою, яка може забезпечити значну ефективність процесу відновлення втрачених блоків на обчислювальній платформі кінцевого користувача.

3.1. Аналіз факторів технологічної ефективності у процесі відновлення втраченої інформації

Технологічні прояви практичної реалізації методів, запропонованих у попередньому розділі, значною мірою впливають на їх ефективність в питаннях реконструювання втрачених блоків даних при їх передачі через глобальну мережу.

Базовими критеріями ефективності механізму відновлення блоків інформації при фіксуванні функціональних параметрів на етапі технічної реалізації є:

- часові затрати на відновлення втрачених блоків інформації;

- обсяги обчислювальних ресурсів, необхідних для вирішення задачі відновлення втрачених блоків;
- обсяг ресурсів мережі, необхідних для ефективного відновлення;
- обсяги використаної пам’яті для виконання функцій відновлення втраченої інформації.

Існує, втім, помітна відмінність між відновленням втрачених блоків при їх зберіганні на ремоуті та технологічними процесами реконструювання даних із використанням завадостійких кодів.

Вагома відмінність технологій реконструювання інформації, запропонованих у даній роботі, від таких, що використовуються в контексті корекції інформації із використанням завадостійких кодів є в тому, що лише вирішення лінійного рівняння не відновлює дані, а тільки описує математичним шляхом процес відновлення даних. Сам процес реконструювання залежить від передачі мережею блоків даних, необхідних для відновлення загублених блоків, їх подальшу буферизацію та виконання обчислювальних операцій пов’язаних із безпосереднім відновленням інформації, втраченої при передачі.

Тому технології, відомі сьогодні, пов’язані із відновленням загублених блоків, що базуються на коригуючих кодах, допускають імовірність попередніх обчислень коригуючого коду для різноманітних похибок блоку даних. Стандартна таблиця розташування зазвичай є оформленням таких обчислень.

Завдання реконструювання блоків інформації, що передається через мережу Інтернет, не уможливлює подібного рішення задачі відтворення. Інакше кажучи, вагомою відмінністю описаної задачі відновлення загублених блоків при їх передачі через глобальну мережу від звичайної корекції похибок є в тому, що безпосереднє відновлення втрачених блоків не використовує матриці, яка визначає лише процедуру відновлення.

Процес реконструювання загублених блоків з технологічної точки зору є розв’язанням системи лінійних рівнянь, коефіцієнти якої сформовані елементами під матриці θ, що є ортогональною матриці A. Також стовпці

ортогональної матриці сформовані із елементів стовпців матриці A, що співвідносяться із загубленими основними блоками при їх передачі, а її рядки сформовані частинами рядків, які співвідносяться із резервними блоками, не загубленими при передачі.

Навіть за умови втрати кількості інформаційних блоків в стовпцях матриці A, що є критичною для відновлювальної спроможності h, може існувати декілька ортогональних під матриць θ.

Зокрема, у випадку, коли кількість інформаційних блоків n рівна дванадцяти (n = 12), і число блоків, які можна гарантовано відновити (h = 4) – число резервних блоків становить m = 10. Тоді матриця A формування резервних блоків, що забезпечує відновлення чотирьох інформаційних блоків, має такий вигляд:

$$A = \begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{vmatrix} \quad (3.1)$$

Якщо припустимо втрату перших двох та двох останніх інформаційних блоків, то в під матриці A, утвореної її 2-ма першими і 2-ма останніми стовпчиками, можливо виявити декілька ортогональних субматриць, в яких є по 4 рядки і 4 стовпці. Так, перші 4 рядки субматриці А формують ортогональну субматрицю θ такого вигляду:

$$\theta_1 = \begin{vmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{vmatrix}$$

Розв’язання такої системи лінійних булеанових рівнянь приводить до отримання виразів реконструювання загублених блоків даних:

$$B_1 = R_1 \oplus R_2 \oplus R_3 \oplus R_4 \oplus B_6 \oplus B_7 \oplus B_8 \oplus B_{10}$$

$$B_2 = R_2 \oplus R_3 \oplus R_4 \oplus B_4 \oplus B_6 \oplus B_7 \oplus B_{10}$$

$$B_{11} = R_1 \oplus R_2 \oplus R_3 \oplus B_3 \oplus B_5 \oplus B_7 \oplus B_8$$

$$B_{12} = R_1 \oplus R_3 \oplus R_4 \oplus B_3 \oplus B_5 \oplus B_8 \oplus B_{10}$$

Зважаючи на виведені формули робимо висновок, що кількість операцій логічних додавань, потрібних для реконструювання 5-ти втрачених блоків, рівна 25-и. Число резервних та інформаційних блоків, використовуваних для реконструювання загублених чотирьох інформаційних блоків дорівнює одинадцяти: чотири резервних $R_1R_2R_3R_4$ та сім основних B_3-B_8B_{10}. А це значить, що задля відновлення загублених блоків необхідно використати 11 невтрачених блоків.

При виборі іншої ортогональної системи, як то ,що сформована 2-им, 3-ім, 6-им та 9-им рядками субматриці A:

$$\theta_2 = \begin{vmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

то вирішення такої системи лінійних рівнянь дозволить описати відмінний математичний вираз реконструкції загублених блоків $B_1B_2B_{11}B_{12}$:

$$B_1 = R_2 \oplus R_9 \oplus B_5 \oplus B_7 \oplus B_{10}$$

$$B_2 = R_6 \oplus R_9 \oplus B_5 \oplus B_6 \oplus B_7$$

$$B_{11} = R_3 \oplus R_6 \oplus B_4 \oplus B_6$$

$$B_{12} = R_9 \oplus B_5 \oplus B_6 \oplus B_8 \oplus B_{10}$$

Стає очевидним факт необхідності 15 операцій логічного додавання наведеної обчислювальної реалізації, що є на шістдесят сім відсотків менше за використання субматриці θ_1. Задля реконструювання загублених блоків $B_1B_2B_{11}B_{12}$ в такому випадку необхідно лише десять блоків, а саме чотири резервних $R_2R_3R_6R_9$ та шість основних $B_4B_5B_6B_7B_8B_{10}$.

Із прикладу, наведеного вище, робимо висновок, що реконструкція загублених блоків може бути здійснена різними способами, що потребують кардинально відмінних обсягів обчислень для вирішення проблеми відновлення.

Максимальні запаси для пришвидшення відтворення загублених блоків даних за використання такої форми відтворення можуть бути доставлені за менший час. Такий спосіб вимагає меншого обсягу блоків, успішно доставлених на сторону приймача, а також вибору форми відтворення, що використовує блоки із максимальною доступністю, а саме, із урахуванням завантаженості вузлів зберігання і поточного трафіку мережі Інтернет.

Практичне просте використання першого із вищеописаних способів можна реалізувати шляхом вибору способу відновлення, в рамках якого система лінійних рівнянь має мінімальну кількість блоків даних, потрібних для відновлення певної комбінації загублених блоків даних. Реалізація 2-го резерву пришвидшення з технологічної точки зору є помітно складнішою за рахунок потреби в постійному моніторингу трафіку мережі Інтернет.

Зважаючи на вищесказане, в ролі основоположного чинника в умовах відтворення блоків даних, загублених під час їх трансферу через мережу Інтернет, увесь набір даних присутній на стороні приймальника, а отже час відновлення залежить від кількості блоків, що при цьому використовуються, тому що час виконання обрахунків є пропорційним зазначеній кількості.

Втім, використання найменшої кількості блоків задля відтворення загублених блоків уможливлює мінімальну кількість обчислювальних операцій логічних додавань для відтворення напряму загубленої чи затриманої на довгий (критичний) час інформації. Отже, висока швидкість процесу відтворення втрачених блоків даних за рахунок обрання оптимального з точки зору мінімальної кількості потрібних для цього кроків, способу відтворення, використання попередніх обрахувань забезпечені прийнятою в основі розробки стратегією використання мінімальної кількості вчасно отриманих блоків.

3.2. Розробка методу прискореного відновлення втрачених блоків інформації за допомогою завчасно утворених таблиць-специфікацій

З точки зору математичної проблеми процес відтворення втрачених блоків m полягає у виокремленні ортогональної субматриці θ розміру $m*m$, стовпці котрої співвідносяться із загубленими блоками даних, з матриці A. Утворена матриця представляє собою систему рівнянь, розв'язок котрої є відновленням загублених блоків даних. Як зазначено вище, можливі різні способи виокремлення на оригінальній матриці A субматриці ортогональної, що відрізняються один від одного кількістю незагублених блоків, дані з яких напряму використовуються для відтворення загублених блоків. Наявність ортогональної субматриці θ у трансформованій матриці Λ теоретично є умовою відтворення загублених блоків у процесі трансферу h основних блоків, номери котрих формують множину Ω.

Як наслідок, з технологічної точки зору процес відтворення загублених блоків даних полягає у розв'язанні систем рівнянь, коефіцієнти котрої сформовані частинами матриці θ. Подальший аналіз матриці трансформованої Λ на предмет знаходження в ній матриці ортогональної θ, на основі матриці θ подальше формування систем рівнянь з їх наступним розв'язком вимагає деяких обчислювальних та часових затрат, які є критичними в питанні відновлення загублених блоків інформації в реальному часу.

Кількість способів відновлення q втрачених блоків інформації h залежить від числа n, що є загальною кількістю інформаційних блоків, а також кількості резервних блоків m, використовуваних для відновлення. Кількість способів реконструювання загублених блоків зростає помітно у випадку, коли кількість реконструйованих блоків значно менша за граничне значення.

Так, для матриці орієнтованої на безумовне відтворення не більше чотирьох загублених блоків даних з дванадцяти $n = 12$ за використання десяти резервних блоків $m = 10$ за відтворення чотирьох загублених блоків даних $h = 4$ існує шість $q = 6$ варіантів відтворення, що вимагають використання від десяти

до дванадцяти неушкоджених блоків та від п’ятнадцяти до двадцяти восьми обчислювальних кроків логічного додавання. За умови використання такої ж матриці для відтворення трьох втрачених блоків існує двадцять чотири способи відновлення $q = 24$, що потребують від восьми до дванадцяти неушкоджених блоків та від шести до шістнадцяти обчислювальних кроків логічного додавання блоків. За умови відтворення пари загублених блоків із використанням матриці існує три способи відтворення, у яких число блоків необхідних для відновлення варіюється від десяти до дванадцяти, а кількість операцій — від дев’яти до тринадцяти. Для відтворення одного втраченого блоку необхідно використати вісім блоків та виконати шість операцій логічного сумування блоків, що були вчасно доставлені.

Висновки до розділу 3

В третьому розділі магістерської дисертації описано технологічні аспекти вирішення систем лінійних рівнянь в контексті опису процедури відтворення втрачених даних. Описані базові критерії ефективності механізму відтворення втрачених блоків інформації, такі як часові витрати, об’єм обчислювальних та мережевих ресурсів, обсяги вживаної пам’яті.

Зважаючи на наведені приклади робимо висновки:

1. Відновлення втрачених блоків інформації є можливим різними шляхами, кожен з яких має свої характеристики витрат часу, пам’яті тощо.
2. Описаний метод пришвидшеного відтворення загублених блоків даних із використанням завчасно утворених таблиць-специфікацій.

РОЗДІЛ 4

РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ЗАПРОПОНОВАНОГО МЕТОДУ ТА МОДЕЛЮВАННЯ

Мета розробки програми – імплементувати запропонований метод для утворення надлишкових лінійних кодів, щоб отримати максимальну вірогідність у відновленні інформації, що була втрачена або спотворена в процесі пересилки через мережу Інтернет. За допомогою програми можливо зробити модель розробленого методу, протестувати та емпірично підтвердити теоретично доведені твердження.

4.1. Опис структури організації даних та змінних

Програмне забезпечення виконано на мові програмуванні C++. Це обумовлено наступними факторами:

- наявність засобів та інструментів, таких структур даних як список, вектор, які зручно використовувати для поставленої задачі моделювання;
- надання можливості написання універсального алгоритму, що буде працювати для різних типів даних;
- надання можливості написання низькорівневих програм, роботи безпосередньо з пам’яттю.

Програма містить один файл з сорс кодом *Program.cpp*.

Для імплементації необхідно було підключити відповідні бібліотеки:

- `<iostream>` – організації *input/output*;
- `<vector>` – надає можливість застосування *vector*;
- `<ctime>` – робота з часом системного годинника;
- `<cstdlib>` – з цієї бібліотеки застосовується метод *rand()* для генерації рандомних значень, що утворюють додаткові вектори;

- `<bitset>` – використовується, щоб вивести результат у бінарному представленні.

У програмі *vector* застосовується, щоб сформувати вхідну систему, що матиме складовими основні та додаткові вектори. Для маніпуляції цими даними викликаються бібліотечні методи класу *vector*.

У реалізації наступні константи оголошені:

- `const int DIGIT` – число бітів, що мають вектори; одночасно відповідають числу основних векторів для системи;
- `const int LIMIT` – значення для обмеження генерації рандомних чисел;
- `const int EXTRA` – число додаткових векторів;
- `const int #Extra` – позиція додаткового вектору для системи, де # може набувати `{First, Second, Third, Fourth, Fifth}`;
- `const int REPEAT` – число повторної генерації рандомних значень для додаткових векторів.

3.2. Організація реалізація методів та функцій

Імплементація включає наступні методи та функції:

- `main()` – головна функція, складовими якої є основні кроки імплементації, викликає інші допоміжні функції та методи, відповідає за генерацію вхідних даних та вивід результата;
- `generateBaseVectors()` – функція, що імплементує процес генерації основних векторів;
- `outputVector()` – функція, за допомогою якої виконується вивід системи векторів;
- `ORT()` – функція, з використанням якої можна визначити факт ортогональності системи.

Програмно ввести вхідні дані можливо такими способами:

- вибрати, щоб додаткові вектори автозгенерувалися;
- в програмі вказати бажане значення для додаткових векторів.

Кількість вкладених циклів, що міститься в імплементації залежить від числа помилок, за яких визначається вірогідність відновлення втрачених даних.

Більше того, для визначення ортогональності системи, необхідно, щоб вона представлялася у вигляді квадратної матриці *n*n* і має ідентичне число рядків та колонок. В імплементації з цією метою додана вкладеність циклів для перебору всіх можливих комбінацій векторів системи.

Після виконання результатом виводу на консоль буде:

- при безпосередньому вводу – вірогідність відновити помилки у ситуації, коли вказані додаткові вектори, відоме число виправлень та сумарне число ситуацій;
- при рандомній генерації – виводиться максимальне та усереднене значення вірогідностей виправлення помилок. Число ітерацій при генерації рівне 10_000.

Наприклад, при кількості основних блоків $n = 64$, додаткових блоків $m = 3$ та втрачених/пошкоджених блоків $e = 3$ код програми буде наступним:

```
#include <iostream>
#include <cstdlib>
#include <ctime>

const unsigned int OPTIMAL = 7;
const unsigned int BASIC = 64;
const unsigned int EXTRA = 3;
const unsigned int ERROR = 3;
//const unsigned int base[] = { 3, 5, 6, 7, 4, 2, 1 };
//for random
unsigned int base[7];

unsigned int baseMatrix[EXTRA][BASIC];

int combination(int, int);

int main()
{
	// random generation
	srand(time(NULL));
	unsigned int random;
	for (int i = 0; i < 7; ++i)
	{
		base[i] = rand() % 7 + 1;
		std::cout << base[i] << " ";
	}
	std::cout << std::endl;
	//generate base matrix
	int h = 1;
```

```
	for (int j = 0; j < BASIC; ++j)
	{
		h = 1;
		for (int i = 0; i < EXTRA; ++i)
		{
			baseMatrix[i][j] = (base[j % OPTIMAL] & h) / h;
			h <<= 1;
		}
	}

	// output matrix
	for (int i = 0; i < EXTRA; ++i)
	{
		for (int j = 0; j < BASIC; ++j)
		{
			std::cout << baseMatrix[i][j] << " ";
		}
		std::cout << std::endl;
	}

	// 3 mistakes
	int count = 0;
	// 3 in basic packets
	for (int i = 0; i < BASIC - 2; ++i)
		for (int j = i + 1; j < BASIC - 1; ++j)
			for (int k = j + 1; k < BASIC; ++k)
			{
				unsigned int checkedVector[3];
				for (int e = 0; e < EXTRA; ++e)
					checkedVector[e] = baseMatrix[e][i] * 4 +
baseMatrix[e][j] * 2 + baseMatrix[e][k];

				unsigned int xor = checkedVector[0] ^ checkedVector[1] ^
checkedVector[2];
				if (xor)
					++count;
			}
	std::cout << "\ncount " << count;
	// 2 in basic and 1 in reserve packets
	unsigned int c;
	for (int i = 0; i < EXTRA; ++i)
		for (int j = 0; j < BASIC - 1; ++j)
			for (int k = j + 1; k < BASIC; ++k)
			{
				unsigned int checkedVector[2];
				c = 0;
				for (int e = 0; e < EXTRA; ++e)
					if (i != e)
						checkedVector[c++] = baseMatrix[e][j] * 2 +
baseMatrix[e][k];

				unsigned int xor = checkedVector[0] ^ checkedVector[1];

				if (xor)
					++count;

			}
	std::cout << "\ncount " << count;
	// 1 in basic and 2 in reserve packets
	for (int i = 0; i < EXTRA; ++i)
		for (int j = 0; j < BASIC; ++j)
```

```
                if (baseMatrix[i][j])
                        ++count;
        std::cout << "\ncount " << count;
        // 3 in reserve packets, calculate as combination
        count += combination(ERROR, EXTRA);
        std::cout << "\ncount " << count;

        int general = combination(ERROR, BASIC + EXTRA);

        float probability = (float)count / general;
        std::cout << "\nprobability " << probability;

        getchar();
        getchar();

        return 0;
}

int combination(int upper, int lower)
{
        int copyUpper = upper;
        int copyLower = lower;
        int dividend = 1;
        int divisor = 1;

        while (copyUpper > 0)
        {
                dividend *= copyLower;
                divisor *= copyUpper;
                --copyLower;
                --copyUpper;
        }
        return dividend / divisor;
}
```

3.4. Інструкція для користувача

Для простоти візьмемо випадок виправлення трьох помилок при наявності в системі трьох додаткових векторів.

Тоді кроки, щоб отримати результат, включають наступний перелік:

1. вписати значення для BASIC – необхідне число основних векторів;
2. вписати значення для EXTRA – необхідне число основних векторів;
3. вписати значення для REPEAT – необхідне число повторів при рандомній генерації;
4. внести потрібне число додаткових векторів, що є константними.

Після внесення всіх необхідних вхідних даних будемо мати

```
const int DIGIT = 3;
const int EXTRA = 3;
const int LIMIT = 0x8;
const int FirstExtra = DIGIT;
const int SecondExtra = DIGIT + 1;
const int ThirdExtra = DIGIT + 2;
const int REPEAT = 10000;
```

Наступними кроками є:

1. потрібно внести значення додаткових векторів у призначеному місці:

```
base = generateBaseVectors(DIGIT);
base.push_back(0x7);
base.push_back(0x6);
base.push_back(0x5);
```

2. відповідно до вхідних даних використати потрібну вкладеність у циклах;
3. скомпілювати та запустити програму на виконання;
4. у результаті у консолі надасться вибір між ручною та рандомною генерацією (рис. 4.1):

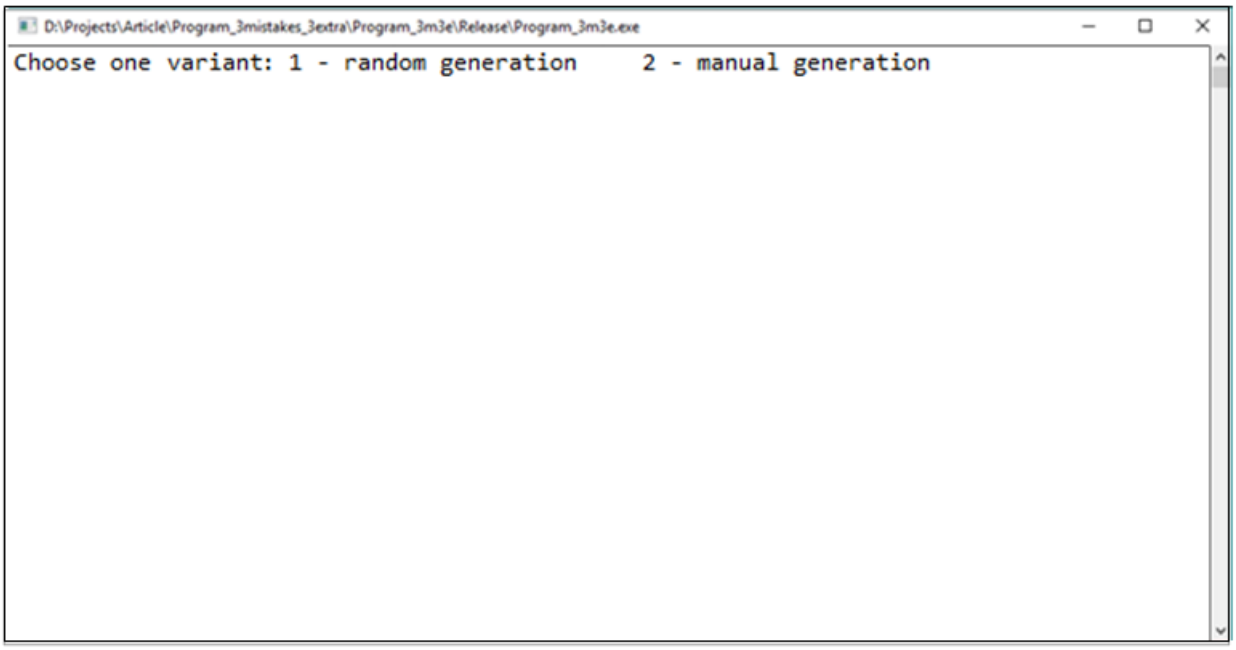

Рис. 4.1. Консоль після компіляції та старту програми

5. вибрати необхідний варіант: рантомне автогенерування натиснувши «1» або ручний ввід – «2»;
6. 1) Обравши перший варіант, автогенерування, на консолі отримаємо (рис. 4.2):

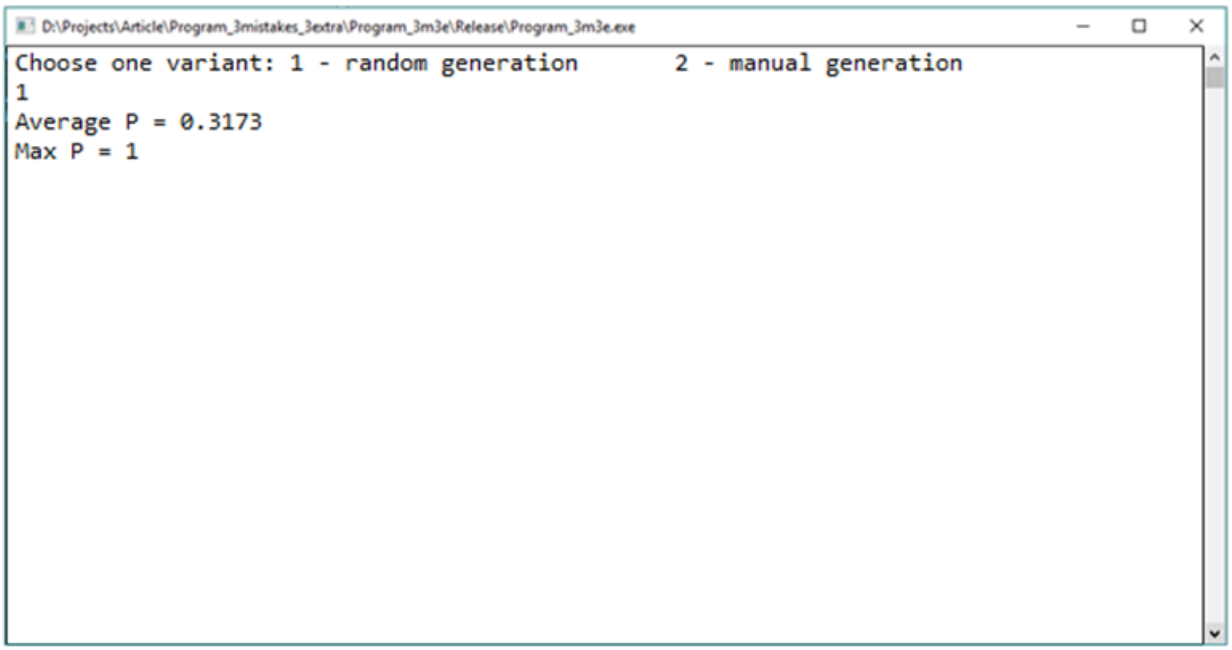

Рис. 4.2. Консоль, якщо обрати автогенерування рандомних додаткових векторів

6. 2) Обравши другий варіант, ручний ввід вхідних даних, на консолі отримаємо (рис. 4.3):

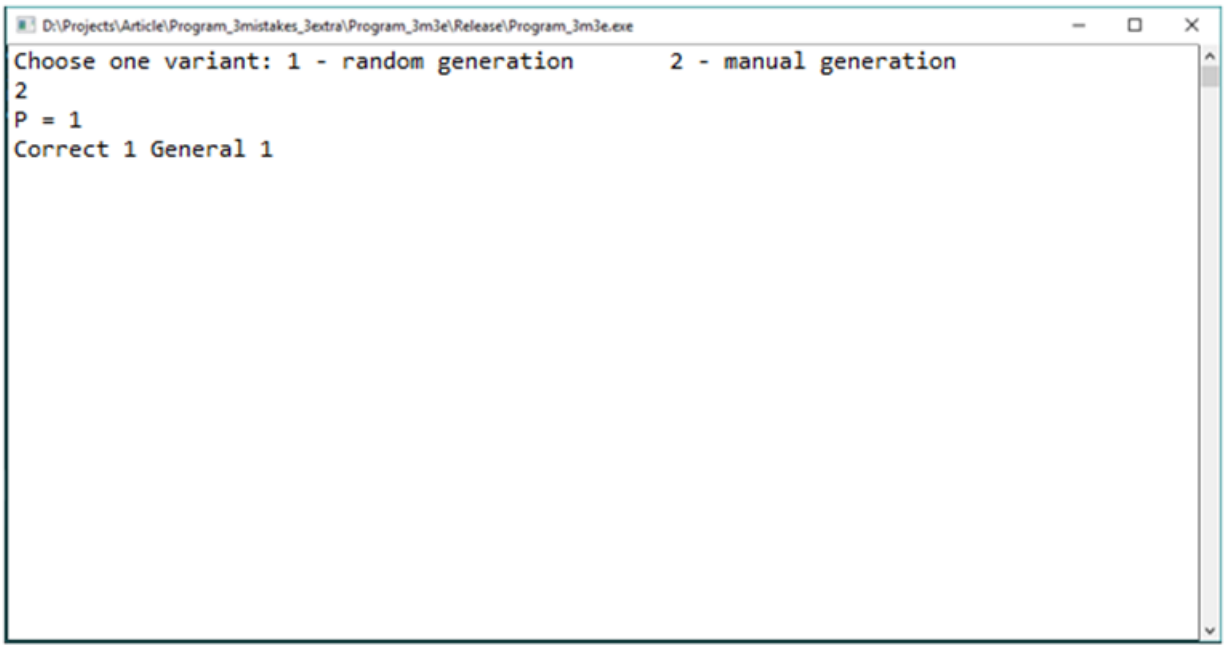

Рис. 4.3. Консоль, якщо обрати ручний ввід вхідних даних для додаткових векторів

Також передбачено випадок неправильного вибору користувачем, тобто будь-який ввід, крім значень «1» та «2», тоді користувач отримує відповідне повідомлення про неправильні введені вхідні дані та можливість повторно зробити вибір (рис. 4.4).

Рис. 4.4. Консоль, якщо користувач надасть неправильні вхідні дані

Висновки до розділу 4

Розділ присвячений описові головних компонентів розробленої програми, метою якої є змоделювати та протестувати запропонований у роботі метод. Узагальнюючи можна зробити такі висновки:

1. Виконано імплементацію програми для моделювання запропонованого методу. Програма виконана з використанням мови програмуванні C++. Вибір мови програмування обумовлений засобами та інструментами, що вона надає, що дозволяє досягти бажану ефективність роботи програмного забезпечення.
2. Для експериментального підтвердження теоретичних даних у реалізації передбачений вибір (рандомним чином або із використанням запропонованого методу) для утворення блоків резервного копіювання.
3. При розробці програмної реалізації у неї були включені можливі випадки локалізації для систем, що містять 3-5 резервні блоки та 3-10 основних блоків, при виникненні різної кількості помилок. Як результат після виконання програми отримуємо ймовірність, що втрачені дані можливо відновити, враховуючи вхідні значення таких параметрів: кількість основних та резервних блоків, кількість помилок, що виникають при передачі у глобальній мережі.

ВИСНОВКИ

В магістерській дисертації виконано теоретичне обґрунтування і одержано нове вирішення наукової задачі підвищення ефективності резервування та відновлення даних в розподілених системах зберігання інформації. Основні наукові і практичні результати полягають у наступному:

1. Виконано аналіз сучасного стану та перспектив розвитку технологій розподіленого віддаленого зберігання інформації, обґрунтовано критерії ефективності засобів резервування та відновлення втрачених даних в віддалених сховищах. Виявлено, що дієвим резервом підвищення ефективності резервування та відновлення даних є урахування конкретних вимог користувачів та особливостей організації віддаленого зберігання даних в реальних системах.

2. Теоретично обґрунтовано, розроблено та досліджено метод відновлення втрачених при віддаленому зберіганні даних, який відрізняється від відомих тим, що дозволяє враховувати інформаційну важливість окремих блоків даних шляхом побудови матриці формування резервних блоків з використанням пріоритетів, що забезпечує підвищення ймовірності відновлення більш важливих блоків при зменшенні ймовірності реконструювання менш важливих за умови, що кількість втрачених блоків перевищує поріг гарантованого відновлення.

3. Розроблено процедуру прискореного відновлення втрачених при віддаленому зберіганні блоків даних шляхом використання спеціальних таблиць, які для кожного варіанту втрат інформаційних чи резервних блоків містять формулу оптимального варіанту реконструювання втрачених інформаційних блоків.

4. Розроблено програмні засоби для реалізації синтезу матриць формування резервних блоків за розробленим методом, а також алгоритми та програми для побудови таблиць для швидкого відновлення втрачених інформаційних блоків. З використанням розроблених програмних засобів проведене експериментальне дослідження ефективності запропонованого

методу резервування та відновлення даних шляхом його імітаційного та статистичного моделювання.

5. Теоретично та експериментально виконано дослідження ефективності запропонованого методу пріоритетного відновлення блоків. Показано, що при втраті блоків, кількість яких перевищує гарантований рівень відновлення, пріоритетні блоки відновлюються з ймовірностями, які на 6-8% більші в порівнянні непріоритетними блоками.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Стіренко С.Г. Забезпечення безперервного відтворення потокового відео в однорангових мережах з використанням erasures кодів / С.Г. Стіренко, А.В. Габінет, Ю.В. Костенко // Вісник НТУУ "КПІ". Інформатика, управління та обчислювальна техніка: збірник наукових праць. – К.: "Век+", 2015. – № 62. – С. 105–110.

2. Таненбаум Э. Компьютерные сети / Э. Таненбаум // СПб: Питер. – 2018. – 960 с.

3. Иванов Д.Г. Метод восстановления данных на основе скошенных матриц в системах распределенного хранения / Луцкий Г.М., Иванов Д.Г. // Известия высших учебных заведений. Проблемы полиграфии и издательского дела. Информационные технологии. – М.: УПИПК МГУП им. И.Федорова. – 2013 – № 2. – С.47-52.

4. Иванов Д.Г. Організація резервування в системах розподіленого зберігання даних // Матеріали 11-ї Міжнародної науково-практичної конференції “Системний аналіз та інформаційні технології САІТ-2013”. Київ, 27-31 травня 2013. – К.: ННК “ІПСА” НТУУ “КПІ”. – 2013. – С.312.

5. Tianli Zhou Fast Erasure Coding for Data Storage: A Comprehensive Study of the Acceleration Techniques / Tianli Zhou, Chao Tian // ACM Transactions on Storage. – March 2020. - Article No.: 7.

6. Мазурков М.И., Герасименко М.Ю. Ортогональные коды на основе совершенных двоичных решеток // Праці УНДІРТ. – 2004. – № 1. – С.13-16.

7. Ivanov D. Paladin: Secure and redundant cloud storage / Heuert U., Ivanov D. // Herald of the Merseburg University of Applied Sciences. – Merseburg: Elbe Drucketei Wittenberg GmbH. – 2011. – № 8. – S.220-228.

8. Corbett P., English D., Goel A., Grcanac T., Kleiman S., Leong D., and Sankar S. Row-diagonal parity for double disk failure. // In Proceedings of the Third USENIX Conference on File and Storage Technologies, USA. – 2004, – pp. 1–14.

9. Иванов Д. Г. Организация резервирования в системах распределенного хранения данных // Вісник Національного технічного університету України ”КПІ” Інформатика, управління та обчислювальна техніка, – Київ: ВЕК+ – 2012. – № 56. – С.160-164.

10. Mortuza A., Kuijper M. A Parametric Approach to List Decoding of Reed-Solomon Codes Using Interpolation // IEEE Transactions on information theory. – Vol.57. – № 10. – 2011. – P.6718-6728.

11. Николайчук Я.М. Коди полів Галуа: теорія і застосування. Тернопіль. – Вид-во ТНУ. – 2012. – 576 с.

12. Nicolae, B., Cappello, F.: BlobCR: Efficient Checkpoint-Restart for HPC Applications on IaaS Clouds using Virtual Disk Image Snapshots. // Proceeding of 24th International Conference for High Performance Computing, Networking, Storage and Analysis. Seattle, USA. – 2011. – pp. 34:1–34:12.

13. Bardis N., Markovskyi O. A Method for Cloud Storage Data Recovery with Limited Loss of Acces // Proceeding of 4-th International Conference on Mathematics and Computers in Sciences and Industry MCSI-17. 24-26 August 2017, Corfu, Greece. P.55-61.

14. Czap L. Secure Network Coding with Erasures and Feedback / L. Czap, C. Fragouli, V. Phabhakaran, S. Diggavi // IEEE Transaction on Information Theory. – 2015. – Vol. 61. – No. 4. – P. 1667-1686.

15. Mladenov T. Raptor Codes for P2P Streaming / T. Mladenov, U. Krieger // Parallel, Distributed and Network-Based Processing . – Feb. 2012, P. 327 – 332.

16. Adler N. Burst-Erasure Correcting Codes with Optimal Average Delay / A. Nitzan., Y.Cassuto // IEEE Transaction on Information Theory. – 2017. – Vol. 63. – No. 5. – P. 2848-2865.

17. Fan X. Variable Packet-Error Coding / X.Fan, O.Kosut, A.B. Wagner // IEEE Transaction on Information Theory. – 2018. – Vol. 64. – No. 3. – P. 1530-1547.

18. Wing Q. End-to-End Error-Correcting Codes on Networks with Wors-Case Bit Errors / Q. Wing, S. Jaggi // IEEE Transaction on Information Theory. – 2018. – Vol. 64. – No. 6. – P. 4467-4479.

19. Leong D. Erasure coding for real-time streaming / D. Leong, T. Ho // Proceedng IEEE Int. Symposium Information Theory. – ISIT-2012. – 200.

20. Cisco Visual Networking Index: Forecast and methodology 2014-2019 [Електронний ресурс] // – Режим доступу: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf

21. The statistics portal [Електронний ресурс] // – Режим доступа: http://www.statista.com/statistics/272835/share-of-internet-users-who-watch-online-videos

22. IEEE 802.3 Industry Connections Ethernet Bandwidth Assessment [Електронний ресурс] // – Режим доступу: http://www.ieee802.org/3/ad_hoc/bwa/BWA_Report.pdf

23. Deshpande H. Streaming live media over a peer-to-peer network / H. Deshpande, M. Bawa, H. Garcia-Molina // In Work at CS-Stanford. – 2002.

24. Erasure coding for cold storage [Електронний ресурс] // – Режим доступу: https://hub.packtpub.com/erasure-coding-cold-storage/

25. Rashmi Vinayak Erasure Coding for Big-data Systems: Theory and Practice / Rashmi Vinayak // Technical Report No. UCB/EECS-2016-155 http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-155.html. – September 14, 2016.

26. Begen A. RFC 7198 – Duplicating RTP Streams / A. Begen, C. Perkins // IETF. – April 2014.

27. Gomez, L.A.B., Distributed diskless checkpoint for large scale systems. / L.A.B. Gomez, N. Maruyama, F. Cappello, S. Matsuoka // Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing. CCGRID ’10, IEEE Computer Society, Melbourne, Australia. – 2010. – P. 63–72.

28. Yuchong H. Proxy-Assisted Regenerating Codes With Uncoded Repair for Distributed Storage Systems. / H. Yuchong, P.C. Patric, W.S. Kenneth // IEEE Transaction on Information Theory. – 2018. – Vol. 64. – No. 4. – P. 2512-2527.

29. Танненбаум,Э. Распределенные системы. Принципы и парадигмы / Э. Танненбаум, М. Ван Стеен. – СПб. : Питер. 2003. - 877 с.

30. Youssefi A.E., Alageel V. A Framework for Secure Cloud Computing // IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 3, July 2012. – PP.487-500.

31. Hu Y. Cooperative recovery of distributed storage systems from multiple losses with network coding./ Y. Hu, Y. Xu, X. Wang., C. Zhan, P. Li // IEEE Journal Sel/Areas Commun. – 2010. –Vol.28. – No.2, - P.268-27.

32. Blaum M. Partial MDS Codes and Their Application to RAID Type of Architectectures / M. Blaum, J. L.Hafner, S.Hetzler // IEEE Transaction on Information Theory. – 2012. – Vol. 59. – No. 7. – P. 4510-4519.

33. Im S. Flash-aware RAID techniques for dependable and high-performance flash memory SSD/ S.Im, D.Shin // IEEE Trans. Comput. –2011. – Vol.C-60, – No.1, – P. 80-92.

34. Li M. C-codes: Cyclic lower-density MDS array codes constructed using starters for RAID-6 / M. Li, J. Shu // IBM New York USA. Res. Report RC25218. – 2011. – 273 P.

35. Lin S. Error Control Codes 2-th edition / S.Lin, J. Shu // NY: Englewood Cliffs NY USA: Prentice-Hall. –2004. – 344 P.

36. Габидулин Э.М. Теория кодов с максимальным ранговым расстоянием // Э.М. Габидулин // Проблемы передачи информацию-1985. – Том.21. – Вып.1. – С.3- 16.

37. Коляда К.В. Метод відновлення даних при їх розподіленому зберіганні на віддалених сховищах / К.В. Коляда, В.О. Романкевич, М.М. Орлова, О.П. Марковський // Комп’ютерно-інтегровані технології: освіта, наука, виробництво. – № 40 – 2020. – С.44-50.

38. Rhea S. Maintence-free global data storage, IEEE Internet Comput. – 2011. – Vol. 5,-No.5, – P. 40-49.

39. Bardis N. Method for Cloud Storage Data Recovery with Limited Loss of Acces. / N. Bardis, O.P. Markovskyi // Proc. of 4-th International Conference on

Mathematics and Computers in Sciences and Industry, MCSI-17, - 24-26 August 2017, Corfu, Greece, – P.55-61.

40. Rashmi K.V. Information-Theoretically Secure Erasure Codes for Distrubute Storage / K.V.Rashmi, B. Nihar, K.Ramchandran // IEEE Transaction on Information Theory. – 2018. – Vol. 64. – No. 3. – P. 1621-1645.

41. Kai X. A Construction of New MDS Symbol-Pair Codes / X.Kai., S.Zhu, P. Li // IEEE Transaction on Information Theory. – 2015. – Vol. 61. – No.11. – P. 5828-5834.

42. Vaushampayan V.A. Reability of Erasure Codes Storage Systems: A Combinatorial-Geometric Approach / V.A. Vaushampayan, A. Campello // IEEE Transaction on Information Theory. – 2015. – Vol. 61. – No.11. – P. 5795-5809.

43. Марковський О.П. Метод резервування та прискореного відновлення даних в системах їх віддаленого зберігання / О.П.Марковський, М.М. Великий // Вісник Національного технічного університету України ”КПІ” Інформатика, управління та обчислювальна техніка, – Київ: ВЕК+ – 2015. – № 63. - С.65-71.

44. Bardis Nikolaos G. Effective method to restore data in distributed data storage systems / Nikolaos G. Bardis, Nikolaos Doukas, Oleksandr P. Markovskyi. // Proceeding of International Conference MILCOM-15, 14-20 Oct.-2015. – USA. P.1101-1109.

45. Kangwook L. The MDS Queue: Analysing the Latency Performance of Erasure Codes / L.Kangwook, B.S. Nihar, H.Longbo, R. Kannar // IEEE Transaction on Information Theory. – 2017. – Vol. 63. – No.5. – P. 2822-2842.

46. Plank J. S. On Practical Use of LDPC Erasure Codes for Distribute Storage Application: Technical Report UT-CS-03-510.- Department of Computer Science, University of Tennessee. – 2004. – 288 P.

47. Sohn J.Y. Capacity of Clustered Distribured Storage / J.Y. Sohn, B.Choi, S.W. Yoon, J. Moon // IEEE Transaction on Information Theory. – 2019. – Vol. 65. – No. 1. – P. 81-107.

48. Chen H.C.H. NCCloud: A network coding based storage systems in cloud-of-cloud / H.C.C. Chen, Y.Hu, P.P.C. Li, Y. Tang // IEEE Transaction on Computers. – 2014. – Vol.63. – No.1. – P.31-44.

49. Stones R. J. K-Plex 2-Erasure Codes and Blackburn Partial Latin Squares / R. J. Stones // IEEE Transactions on Information Theory. – 2020. – vol. 66. – № 6. – P.3704-3713.

50. Suh Changho. Exact-Repair MDS Code Construction Using Interference Alignment / Changho Suh, Kannan Ramchandran // IEEE Transaction on Information Theory. – 2011. – Vol.57, – No. 3, – PP. 1425-1442.

51. Марковський О.П. Метод резервування та прискореного відновлення даних в системах їх віддаленого зберігання/ О.П.Марковський, Д.Г. Іванов, М.М. Великий, М.В. Невдащенко // Вісник Національного технічного університету України ”КПІ” Інформатика, управління та обчислювальна техніка, – Київ: ВЕК+ – 2014 – № 60. С.46-54.

52. Calder B. Windows Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency / B. Calder // ACM SOSP. – 2011. – № 2. – P.62-70.

53. Lakshman A. Cassandra: A decentralized structured storage system./ A. Lakshman, P, Malik // SIGOPS Oper. Syst. Rev. – 2010. – № 44, – P. 35–40.

54. Blaum M. Array Codes With Local Properties / M. Blaum, S. R. Hetzler // IEEE Transactions on Information Theory. – 2020. – vol. 66. – № 6. – P.3675-3690

55. Ivanov D. Paladin: Secure and redundant cloud storage / Heuert U., Ivanov D. // Herald of the Merseburg University of Applied Sciences. - Merseburg: Elbe Drucketei Wittenberg GmbH. – 2011. – № 8. – S.220-228.

56. Иванов Д. Г. Организация резервирования в системах распределенного хранения данных // Вісник Національного технічного університету України ”КПІ” Інформатика, управління та обчислювальна техніка, – Київ: ВЕК+ – 2012. – № 56. – С.160-164.

57. Ghemawat S. The Google file system / S.Ghemawat, H/Gobioff, S.T. Leunh // Proceeding of ACM Symposium on Operating Systems Principles (SOSP). October 19–22, 2003, Bolton Landing, New York, USA.– 2003. – P.29-43.

58. Тормасов,А. Г. Модель распределенного хранения данных с регулируемой избыточностью / А. Г. Тормасов, М. А. Хасин, Ю. И. Пахомов // Электронный журнал «Исследовано в России». 2001. – С.14-22.

59. Vrable M. Cumulus: Filesystem backup to the cloud / M. Vrable, S. Savage and G. Voelker // ACM Trans. on Storage (ToS), – 2009. – Vol. 54. – № 12, – P.11-17.

60. Fan, B. Diskreduce: Raid for data-intensive scalable computing./ B.Fan, W.Tantisiriroj, L. Xiao, G. Gibson // In: PDSW '09: Proceedings of the 4th AnnualWorkshop on Petascale Data Storage. – 2009. – ACM, Portland, USA. – P. 6–10.

61. Ташков П. А. Восстанавливаем данные на 100 %. - СПб.: Питер, – 2010. – 208 с.

62. Yehezkeally Y. and Schwartz M. Reconstruction Codes for DNA Sequences With Uniform Tandem-Duplication Errors / Y.Yehezkeally, M. Schwartz // IEEE Transaction on Information Theory. – 2020. – Vol. 66. – No. 5. – P. 2658-2669.

63. Doroshenko L. Organization of Reservation and Reconstruction of Data / Doroshenko L., Markovskyi O., Honchar A. // International Conference ICSFTI2019 (Kyiv, May 14−15, 2019). Kyiv, 2019. P. 29-34.

64. Doroshenko L. Organization of Data Recovery when Storing in the "Clounds" / Doroshenko L. // Proceedings of the XXI International Students R&D Online Conference (Kyiv, 17 December, 2020). Kyiv, 2020. P. 29-31.

65. Дорошенко Л.Ю. Метод резервування даних з використанням лінійних відновлюючих кодів / Л.Ю. Дорошенко, В.Ю. Куц // Матеріали II міжнарод. наук.-практ. конф. Наука та концепції: (м. Київ, 29-30 квіт. 2019 р.). Київ, 2019. С. 19-22.

66. Дорошенко Л.Ю. Організація відновлення даних при їх зберіганні в хмарах / Л.Ю. Дорошенко, О.В. Русанова, В.Ю. Куц // Матеріали III міжнарод. наук.-практ. конф. (м. Київ, 14-15 трав. 2020 р.). Київ, 2020. С. 33-37.